Crane Oscillation Control:
Nonlinear Elements and Educational Improvements

A Thesis
Presented to
The Academic Faculty

by

Jason W. Lawrence

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Mechanical Engineering
Georgia Institute of Technology
August 2006

Crane Oscillation Control:
Nonlinear Elements and Educational Improvements

Approved by:

Dr. William Singhose, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Professor Steven Danyluk

Professor Donna Llewellyn
(Enhancement of Teaching and Learning)

Professor Nader Sadegh

Dr. Neil Singer

Date Approved: July 4, 2006

Date Approved _____

To my family,

Austina, Mom, Dad, and Justin,

I would not have come this far if not for you.

ACKNOWLEDGEMENTS

This work would not have been possible without the support of several people and organizations. Siemens donated all of the motors, drives, PLCs, digital cameras, and software. Siemens also funded the graduate students working on project. Invaluable support was provided from Siemens by Christian Callegari, Eddie Prince, and Andrew Miller. Igus donated the cable management parts as well as the bearings for the Tower Crane. The Tokyo Institute of Technology also provided assistance and support for the Tower Crane during its stay in Japan. Whiting Crane Co. helped install the new electrical system in the HiBay crane.

There were also several individuals who worked with me to make these projects possible. On the tower crane, Urs Glauser designed the electrical layout and his students, Adrian Erb and Rolf Weirs, programmed the PLO and GUI interface. In addition, John Hued assisted in programming the vision system and Brian Catkin machined the parts. On the bridge crane Dustbin Bergman machined all of the parts and assisted with the assembly.

Finally, I would like to thank my advisor Dr. Singhose. His unwavering confidence in my work has helped refine the results to a level I would not have thought possible.

TABLE OF CONTENTS

DEDICATION . . . . . **iii**

ACKNOWLEDGEMENTS . . . . . **iv**

LIST OF TABLES . . . . . **x**

LIST OF FIGURES . . . . . **xi**

SUMMARY . . . . . **xvii**

I INTRODUCTION **1**

1.1 Payload Swing and Input Shaping 1

1.2 Problem Description 4

1.3 Goals and Contributions 7

II BACKGROUND **9**

2.1 Cranes and Crane Control 9

2.1.1 Crane Control Methods 11

2.2 Input Shaping Theory 12

2.2.1 Input Shaping Properties 15

2.3 Input Shaping and Cranes 18

2.3.1 Background: Basic Input Shaping for Cranes 19

2.3.2 Input Shaping Variations for Cranes 19

2.4 Input Shaping for Nonlinear Systems 20

2.4.1 Robust Shaping 20

2.4.2 Adaptive Shaping 21

2.4.3 Linearization 21

2.4.4 Combining Input Shaping with Other Control Elements 23

2.4.5 Boundary Value and Numerical Optimization Techniques 24

2.4.6 Closed-Form Solutions 25

2.5 Adaptive Input Shaping 25

III EXPERIMENTAL CRANE DESIGNS **28**

3.1 Overview:Crane Projects 28

3.2 HiBay Crane 30

3.2.1 Electrical Hardware Description 30
3.2.2 PLO programming 33
3.3 Bridge Crane 35
3.3.1 Axes Design 37
3.3.2 Design for Portability 38
3.3.3 Vision System 38
3.4 Tower Crane 40
3.4.1 Graphical User Interface (GUI) 42
3.4.2 Internet Control 43
3.4.3 Hoisting 44
3.4.4 Anti-Sway Controller 44
3.5 Conclusions: Lessons Learned 46
3.5.1 Modeling and Layout: 3-D vs. 2-D Computer Design 46
3.5.2 Portability 47
3.5.3 Material Selection 47
3.5.4 Choice of Actuators 48
IV CURRICULUM DEVELOPMENT 49
4.1 Advances in Dynamics and Controls Curriculum 49
4.1.1 Core Concepts 50
4.1.2 Educational Milestones 50
4.2 Education Background 52
4.3 Phase I: Initial Curriculum Development at Georgia Tech 55
4.3.1 Phase I Conclusions 56
4.4 Phase II: Curriculum Development at GT Lorraine 56
4.4.1 Lab Descriptions 57
4.4.2 Undergraduate Laboratory Course 61
4.4.3 Phase II Conclusions 62
4.5 Phase III: Curriculum Development at Tokyo Tech 62
4.5.1 Laboratory Assignments 63
4.5.2 Labs 1 and 2: Characterizing the Unshaped Response 64
4.5.3 Labs 3 and 4: Input Shaper Design and Implementation 64

4.5.4 Lab 5: Crane Path Planning and Automation 65
4.5.5 Student Final Projects 65
4.5.6 Follow-Up Test 66
4.5.7 Phase III Conclusions 67
4.6 Phase IV: Final Lab Modules 67
4.7 Curriculum Development Conclusions 69
4.7.1 Future Work 69
V CRANE OPERATOR STUDIES 71
5.1 Operator Study Overview 71
5.2 Local: Completion Time 73
5.3 Local: Path Choice 75
5.4 Local: Number of Collisions 75
5.5 Local: Operator Learning 76
5.6 Remote: Operator Performance 77
5.7 Remote vs. Local Operation 79
5.8 Operator Study Conclusions 79
VI COMMAND SHAPING FOR A FLYBALL SYSTEM 82
6.1 Dynamic Analysis of the Flyball System 84
6.1.1 Equilibrium at Constant Angular Velocity 86
6.2 A ZV Shaper for a Partially Linearized Flyball Model 87
6.2.1 ZV_{fly1} Shaped Short Commands 89
6.2.2 Discussion of ZV_{fly1} Assumptions 91
6.2.3 ZV_{fly1} Performance Over a Range of Parameters 93
6.3 A Numerically Derived ZV Shaper for the Full Flyball Model 94
6.3.1 Comparison of ZV_{fly1} and ZV_{fly2} Commands 97
6.4 A Improved ZV Shaper for a Partially Linearized Flyball Model 99
6.5 A UMZV Shaper for a Partially Linearized Flyball model 100
6.5.1 $UMZV_{fly}$ Long Commands 101
6.5.2 $UMZV_{fly}$ Medium Commands 102
6.5.3 $UMZV_{fly}$ Short Commands 104

6.5.4 $UMZV_{fly}$ Command Summary and Performance 106
6.6 Flyball Study Conclusions 110
VII INPUT SHAPING FOR TOWER CRANES 112
7.1 Tower Crane Equations of Motion 112
7.1.1 Derivation of Equations of Motion 115
7.1.2 Comparison of Equations of Motion with Literature 117
7.2 Analysis and Linearization of Tower Crane Equations 118
7.2.1 Radial Motion 118
7.2.2 Rotational Motion 119
7.2.3 Steady State Deflection 119
7.2.4 Linearization 121
7.3 Input Shaping for Tower Cranes 124
7.3.1 Standard Shapers: ZV, UMZV, and ZVD 125
7.3.2 2 Mode ZV Shaper 125
7.3.3 Radial Assisted Shapers 128
7.3.4 Linearized Shapers 134
7.3.5 Simulation Results 155
7.3.6 Tower Crane Shaping: Experimental Results 164
7.4 Tower Crane Study Conclusions 168
VIII INPUT SHAPING FOR DRIVE NONLINEARITIES 171
8.1 Vector Approach for Continuous Systems 175
8.1.1 Single Step Response 175
8.1.2 Multi-Step Response 178
8.1.3 Insight into Vector Response 181
8.1.4 Zero-Vibration Shaper Approach 189
8.1.5 Template Shaper Approach 190
8.1.6 Summary of the Vector Approach for Continuous Systems 195
8.2 Application: Braking Nonlinearity 195
8.2.1 System Definition and Experimental Setup 196
8.2.2 ZV Shaping with a Braking Nonlinearity 197

8.2.3 UMZV Shaping with a Braking Nonlinearity 202
8.2.4 Formulation and Implementation of a $UMZV_C$ Shaped Command. 205
8.2.5 Evaluation of $UMZV_C$ Commands 209
8.2.6 $UMZV_C$ Commands: Effect of Pulse Duration 210
8.2.7 Conclusions 212
8.3 Vector Approach for Digital Systems 212
8.3.1 Digitizing the Vector Solution 214
8.3.2 Digital Analysis Tool 218
8.3.3 Digital Solution Techniques 222
8.4 Application: Simulations of a Randomly Generated Drive nonlinearity . . 227
8.4.1 Simulating the Nonlinear Drive Dynamics with Polynomials 227
8.4.2 Simulation Setup 230
8.4.3 Simulation Results 232
8.5 Application: Low Gain Bridge Crane Experiments 234
8.6 Vector Approach: Conclusions 239
IX CONCLUSIONS 242
9.1 Future Work 243
APPENDIX A — JAPAN/US FINAL PROJECTS 247
APPENDIX B — FINAL LAB MODULES 254
APPENDIX C — $ZV2_{LIN}$ POLYNOMIAL COEFFICIENTS 269
APPENDIX D — $UMZV2_{LIN}$ POLYNOMIAL COEFFICIENTS 271
APPENDIX E — EDUCATION FOLLOW-UP TEST 273
REFERENCES 277

LIST OF TABLES

5.1 Summary of Operator Studies 73

5.2 Simple Obstacle Internet Operator Study. 78

5.3 Crane Operator Summary 81

7.1 Nominal System Parameters. 156

7.2 Average Vibration Reduction Compared to Unshaped Motion. 158

7.3 Comparison of Shaper Vibration Performance 158

7.4 Comparison of Shaper Duration 161

7.5 Comparison of Shaper Robustness to Suspension Length 163

7.6 Comparison of Shaper Robustness to Slew Velocity 163

7.7 Nominal Tower Crane System Parameters. 164

7.8 Experimental Average Vibration Reduction Compared to Unshaped Motion. 166

7.9 Performance of Various Shapers. 169

8.1 Bridge Crane Experimental Parameters. 197

C.1 Coefficients for Rising Portion of the $ZV2_{Lin}$ Command for $L = [10 \ldots 30]$ m and $\dot{s} = [0.05 \ldots 0.1]$ rad/s. 269

C.2 Coefficients for Falling Portion of the $ZV2_{Lin}$ Command for $L = [10 \ldots 30]$ m and $\dot{s} = [0.05 \ldots 0.1]$ rad/s. 270

D.3 Coefficients for Rising Portion of the $UMZV2_{Lin}$ Command for $L = [10 \ldots 30]$ m and $\dot{s} = [0.05 \ldots 0.1]$ rad/s. 271

D.4 Coefficients for Falling Portion of the $UMZV2_{Lin}$ Command for $L = [10 \ldots 30]$ m and $\dot{s} = [0.05 \ldots 0.1]$ rad/s. 272

LIST OF FIGURES

1.1 Two Common Cranes: Tower Cranes and Bridge Cranes. 2
1.2 Transport Operation: Command and Response. 2
1.3 Transport Operation for a Tower Crane. 5
1.4 Braking Nonlinearity Command and Response. 6
1.5 Operator-in-the-Loop Control. 6
2.1 Bridge and Tower Crane Sketches. 10
2.2 Planar Crane Model. 10
2.3 Block Diagram of an Input Shaped Crane System. 13
2.4 Desired and Shaped Velocity Commands. 13
2.5 Generalized Input Shaper. 14
2.6 Superposition of Impulse Responses. 14
2.7 Various Shaped Pulse Commands. 16
2.8 Sensitivity Curves for Various Shapers. 18
3.1 Progression of Crane Projects 29
3.2 HiBay Crane Picture. 31
3.3 HiBay Crane Sketch. 31
3.4 HiBay Crane Electrical Hardware Overview 32
3.5 Real-Time input shaping. 34
3.6 PLO Program Structure 35
3.7 Bridge Crane Experimental Testbed: Sketch and Picture. 36
3.8 Bridge Crane Block Diagram. 36
3.9 Bridge and Trolley Axes. 37
3.10 Bridge Disassembly Picture. 38
3.11 Vision System Designs. 39
3.12 Tower Crane Picture. 41
3.13 Tower Crane Sketch. 41
3.14 Tower Crane System Overview. 42
3.15 Tower Crane Computer Interface. 43
3.16 Cable Spreader on the Hoisting Pulley. 44

3.17 Anti-Sway Feedback Control Loop for Trolley and Slewing Axes. 45
4.1 Time-line of Educational Milestones 51
4.2 Overview of Universities Teaching Input Shaping. 54
4.3 Lab Assignments Overview 57
4.4 Sample Student Project: Pouring Out Sand to Spell "GT". 61
4.5 Student Data from Labs 1 through 4. 64
4.6 Student Data from Labs 5. 65
5.1 Operator-in-the-Loop Control. 71
5.2 Sample Obstacle Course and Operator Response 72
5.3 Overhead Sketch and Run Time Data for Courses 1 and 2. 74
5.4 Path Choice for Courses 1 and 2. 75
5.5 Overhead Sketch and Collision Data for Course 3. 76
5.6 Overhead Sketch and Learning Data for Course 4. 77
5.7 Data from Remote Operator Study. 78
5.8 Overhead Sketch, Run-Time Data, and Collision Data for Courses 5 and 6. 80
6.1 One Mass Flyball System. 82
6.2 Response of the Flyball System to ZV shaped and Unshaped Commands. . 83
6.3 Sketch and Freebody Diagram of Flyball System. 84
6.4 Steady-State Deflection ϕ_{ss} for various parameters. 87
6.5 Forming a ZV Command for the Linearized Flyball Model 88
6.6 ZV_{fly1}, ZV, and Unshaped Simulations [L=1 m, R=0.75 m, $\sigma_f = 0.5$ m/s]. . 89
6.7 Forming a ZV Short Command for the Linearized Flyball Model. 89
6.8 ZV_{fly1}, ZV, and Unshaped Simulations [L=1 m, R=0.75 m, $\sigma_f = 0.5$ m/s]. . 90
6.9 ZV_{fly1}Commands and Responses for Various t_p. 90
6.10 Parameter Space in which Linearizing Assumptions are Valid 92
6.11 Comparison of Residual Vibration of Unshaped and ZV_{fly1} Shaped Commands. 93
6.12 Outline of ZV_{fly2} Algorithm. 95
6.13 "Shooting" Algorithm Command Guesses and Responses. 95
6.14 ZV_{fly1} and ZV_{fly2} Residual Vibration for $\sigma_f = 0.25$ m/s. 98
6.15 Numerical Errors in ZV_{fly2} Solution. 98
6.16 Comparison of ZV_{fly1} and ZV_{fly2} Parameters for L=3 m and R=1 m. 99

6.17 ZV_{fly3} Command and Comparison to Other Commands. 100
6.18 Formation of $UMZV_{fly}$ Command/ 101
6.19 Possible Templates for the $UMZV_{fly}$ medium command. 102
6.20 $UMZV_{fly}$ Medium Command for Various Pulse Times. 104
6.21 Evolution of $UMZV_{fly}$Short Command. 105
6.22 $UMZV_{fly}$ Switch Times for Various t_p. 107
6.23 $UMZV_{fly}$ Move Distance for Various t_p. 107
6.24 Comparison of Residual Vibration of $UMZV_{fly}$ and ZV_{fly1} Commands for $\sigma_f = 0.25$ m/s. 108
6.25 Comparison of Duration of $UMZV_{fly}$and ZV_{fly1}commands for various t_p. . . 109
7.1 Tower Crane Sketch: Side View and Overhead View. 113
7.2 Free-Body Diagram of Payload 114
7.3 Equations of Motion Test: Derived Model vs. Golafshani's Model 118
7.4 Linearized Frequencies for Various Slew Velocities 122
7.5 Constant Slew Velocity Simulation Results [R=1, L=1]. 123
7.6 Generic Shaping Process 125
7.7 Sensitivity Curve for EI Shaper. 126
7.8 ZV2 Shaper Impulses and Vector Diagram. 127
7.9 Relating the Steady State of the Crane to ZV shaping. 129
7.10 Tower Crane Steady State Condition. 130
7.11 Radial Assist Shaper Formation: First Stage. 132
7.12 Radial Assist Shaper Formation: Second Stage. 132
7.13 Radial Assist Shaper Formation: Third Stage. 132
7.14 Limitations of Linear Model. 135
7.15 Physical Configuration during Step Transition. 136
7.16 Response to Multi-Step Slew Command. 140
7.17 Forming Vector Diagram to Multi-Step Slew Command. 142
7.18 $ZV2_{Lin}$ Command Template. 144
7.19 Vector Diagram for $ZV2_{Lin}$Slewing Command. 146
7.20 A_1 values of $ZV2_{Lin}$ for Various Parameters. 147
7.21 $ZV2_{Lin}$ command and response. 148

7.22 UMZV2$_{Lin}$ Shaper Template. 149
7.23 Using Vector Diagrams for UMZV2$_{\text{Lin}}$ Commands. 151
7.24 t_2 values of UMZV2$_{\text{Lin}}$ Command for Various Parameters. 152
7.25 UMZV2$_{\text{Lin}}$ command and response. 154
7.26 Residual Vibration Measurement 157
7.27 Residual Vibration of Shaped and Unshaped Commands for Various Pulse Times. 158
7.28 Shaper Vibration for Various Parameters 160
7.29 Shaper Duration for Various Parameters 161
7.30 Shaper Robustness 162
7.31 Residual Vibration of Standard Shapers for Short Commands 164
7.32 Comparison of UMZV2$_{\text{Lin}}$ Command and Response, With and Without Tower Crane Acceleration Saturation. 165
7.33 Residual Vibration of Various Commands for Different Pulse Times. 166
7.34 Experimental Data: Residual Vibration of Various Shaped Commands for Different Tower Crane Configurations. 168
8.1 Different Representations of Nonlinear Cranes 171
8.2 Example of a Velocity Command and the Nonlinear Trolley Response. . . . 173
8.3 Describing Function Analysis verses Nonlinear Drive Analysis. 174
8.4 Block Diagram and Plant for Single Step Simulation. 175
8.5 Single Step Command and Response. 176
8.6 Plotting the Vibration Vectors. 177
8.7 Block Diagram for input-shaped Crane 178
8.8 Breaking Down Command for Analysis. 179
8.9 Vector Analysis Applied to Three Cases. 182
8.10 Vector Diagram and Frequency Response for Case I. 184
8.11 Relationship between Shaper Response Vectors ($\vec{x}_i$) and Linear Response Vectors ($\vec{y}_i$.) 186
8.12 Frequency Response of Shaper and Linear System. 186
8.13 Relationship between Shaper Response Vectors ($\vec{x}_i$) and Nonlinear Response Vectors ($\vec{v}_i$.) 188
8.14 Frequency Response of Shaper, Linear System, and Nonlinear System. . . . 189

8.15 Frequency Response of Shaper, Linear System, and Zero-Vibration Nonlinear Shaper. 190

8.16 Frequency Response of Nonlinear Shaper (Template Design) for Various Levels Nonlinear Dynamic Behavior. 193

8.17 Non-Symmetrical Acceleration-Braking. 196

8.18 Crane Block Diagram. 197

8.19 Short, Long, and Interference ZV Commands. 199

8.20 Residual Vibration using Various Pulse Durations $[\tau_b = 0.065\ (s)]$. 201

8.21 Vibration of ZV Commands for Various t_p and τ_b. 202

8.22 Deconvolution of a UMZV Shaped *Short Command*. 203

8.23 UMZV_C Command Template. 203

8.24 Vibration of Unshaped and UMZV Shaped Pulse for Various Pulse Durations $[\tau_b = 0.065(s)]$. 205

8.25 Vibration of UMZV Commands for Various t_p and τ_b. 206

8.26 Phaser Diagram. 208

8.27 Constraints on τ_a/T and τ_b/T for a UMZV_C Shaped Command. 211

8.28 UMZV_C Shaped and Unshaped Residual Vibration for Various Pulse Durations $[\tau_b = 0.065(s)]$. 211

8.29 Vibration of UMZV_C Commands for Various t_p and τ_b. 213

8.30 Digitizing the Trolley Velocity. 214

8.31 Original $g_i(k)$ Segment 216

8.32 Analysis of $g_i(t)$ segment. 216

8.33 Vector Diagram Tool for Nonlinear Systems. 219

8.34 Block Diagrams for Forming the Template Shaper Response and Nonlinear Response. 220

8.35 Analysis Tool for Nonlinear Systems. 221

8.36 Example of using the Nonlinear Analysis Tool. 222

8.37 Nonlinear Shaper Performance after 1 iteration. 224

8.38 Forming a Randomly-Generated Drive Nonlinearity. 228

8.39 Nonlinear Drive System Block Diagram. 230

8.40 Responses of Random Drive Nonlinearity Algorithm. 231

8.41 Simulation Overview 232

8.42 Comparison of Standard and Nonlinear Shapers after 5 iterations. 233

8.43 Bridge Crane Low-Gain Experiments: Setup and Experimental Data. 235
8.44 Bridge Crane Response to EI Command. 236
8.45 Experimental Nonlinear Shaper Solution Process. 237
8.46 Standard and Nonlinear Shaper Experimental Results. 238

SUMMARY

Command Generation has been shown to be a practical and effective control scheme for eliminating payload swing on industrial cranes. However this technology has not been used to its full potential. One reason is that nonlinear crane dynamics degrade the performance of current command generators, making them challenging to use. A second reason is that few crane operators are aware of this technology. Therefore, this thesis strives to alleviate these problems through the completion of three major tasks. First, new command generation algorithms are developed that compensate for nonlinear crane dynamics. Two major sources of non-linear dynamics are targeted: nonlinear drive dynamics, and nonlinear physical dynamics of tower cranes. Second, command generation are examined from an educational perspective; both in the classroom and in the working field. Third, three experimental crane devices were built to fulfill the two prior tasks.

CHAPTER I

INTRODUCTION

Cranes have been used for thousands of years to lift and transport heavy materials. They are used in shipyards, construction sites, and warehouses. In all cases the most common operation of a crane is point-to-point transport of a payload. Two common types of cranes are shown in Figures 1.1(a) and 1.1(b), a tower crane and a bridge crane respectively. The bridge crane operates using Cartesian (x-y) axes, while a tower crane uses polar (r-θ) axes.

There are three factors are of paramount importance in crane operations: speed, accuracy, and safety. For instance, on most construction sites, speed is important for keeping costs down. However if the crane is used to hoist a beam into place where it will be pinned to the existing structure, accuracy will be important for aligning the beam. Safety is an issue for all cranes because of the potential for collisions between objects and people.

1.1 Payload Swing and Input Shaping

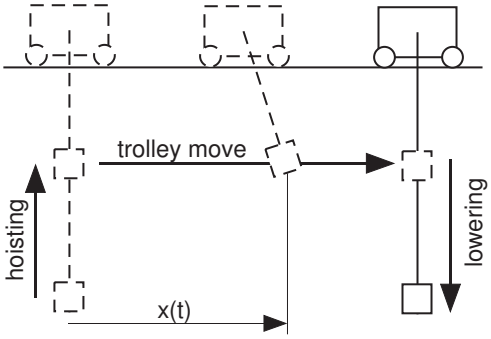
All cranes use cables to hoist and support the payload. Because of this structure the payload has the tendency to swing during transport operations. Swinging degrades the speed, accuracy, and safety of transport operations; the same three factors earlier noted as being of paramount importance to crane operation. It lowers the speed of transport operations because the payload swing must die out before the payload can be safely lowered into position. The swing makes it difficult to perform alignment, fine positioning, or other accuracy driven tasks. Swing also causes safety problems because of the potential for collisions with objects or people.

Input shaping is an easy and effective way to reduce payload swing in cranes. Input shaping works by taking the desired command and convolving it with a sequence of impulses. To see input shaping at work, consider the transport operation shown in Figure 1.2(a). The operation usually consists of a three-step procedure: hoisting the payload to a safe

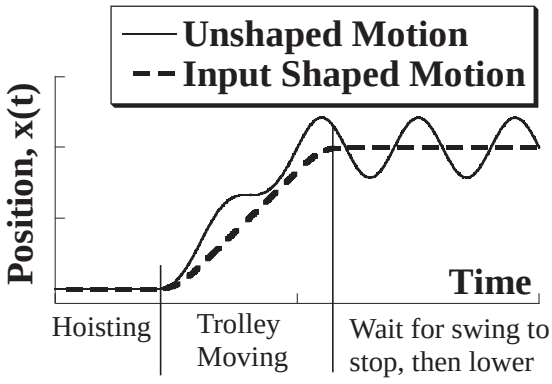
(a) Tower Crane

(b) Bridge Crane

Figure 1.1: Two Common Cranes: Tower Cranes and Bridge Cranes.



(a) Transport Operation Command.



(b) Shaped vs. Unshaped Response

Figure 1.2: Transport Operation: Command and Response.

height, repositioning the payload over the desired location, lowering the payload in position. The horizontal position of the payload can be plotted as a function of time, as shown in Figure 1.2(b). From this graph it is clear that the payload swings during and after the trolley's movement. Now suppose the same operation was performed, but with input shaping implemented. Figure 1.2(b) shows that with input shaping the swing is essentially eliminated. More details on this process will be revealed in chapter 2.

Input shaping has several advantages over other control techniques that make it a very attractive solution for cranes:

- No added sensors are necessary. Many other crane control techniques require realtime feedback of the payload as it swings. Input shaping requires no feedback.

- The input shaper works on-the-fly so the operator is free to drive through any desired path. Most optimal trajectory/control schemes need to know the desired path ahead of time.

- Input shaping is simple and cheap to implement. The algorithm is easy to program into any microcontroller. In contrast, many other crane control schemes are computationally intensive.

- Input shaping is easy for crane operators to use because it has an intuitive feel. When the operator pushes forward, the crane moves forward. When the operator releases the forward button the crane stops. The only difference between shaped and unshaped operation, from the operator's standpoint, is an increase in the response time. Other crane control algorithms can use complicated algorithms that are difficult for crane operators to adapt to.

These strengths are of paramount importance. This thesis will seek new, improved input shaping techniques. Therefore it is important that these new solutions strive to retain the aforementioned strengths or add to them.

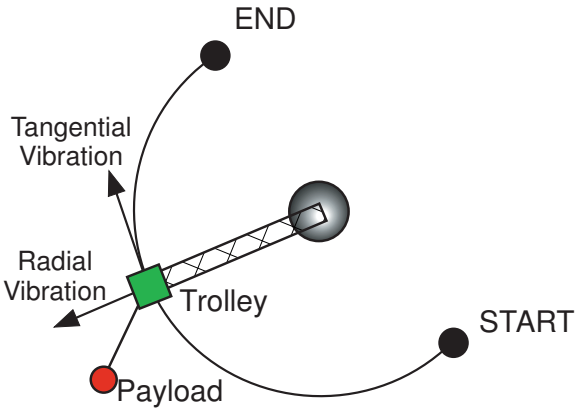
1.2 Problem Description

Although input shaping is a powerful solution for crane control, payload swing continues to be a problem for cranes. One reason is that nonlinear crane dynamics degrade the performance of input shapers. A second reason is that despite the flurry of academic activity, input shaping has not been integrated into the engineering curriculum. As a result the technology remains an idea implemented on only a few commercial cranes. In summary, input shaping is a good technology, but has not unlocked its full potential in the field of cranes or in education.

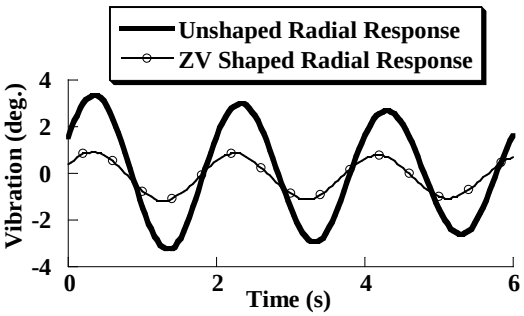
Nonlinear systems are problematic for input shapers because they are designed for linear systems. Nonlinear crane dynamics can come from several sources. This study targets two common sources: nonlinearities in the drive system, and nonlinearities caused by tower crane structures. Nonlinearities in tower cranes come from rotational motion. As the crane rotates it induces centripetal and Coriolis accelerations that make the crane's dynamics nonlinear, even for small swing angles[1]. These nonlinearities degrade an input shapers ability to eliminate payload swing. As an example, consider a single rotational point-to-point motion, as sketched in Figure 1.3(a). The payload swing can be measured in two directions, radial and tangential. Figures 1.3(b) and 1.3(c) show the radial and tangential residual vibration for a 35^0 move using an experimental tower crane in Japan. The crane is 2 meters tall with a workspace 2 meters in diameter, and is equipped with a camera to measure payload vibration. The payload vibrates at the end of the move as expected. The figures also show the residual vibration using an input shaper. The input shaper clearly reduces the vibration, but does *not* eliminate it due to the nonlinearities of the motion. This motivates the question: *Can an input shaper be designed to eliminate vibration in tower cranes?*

A second source of nonlinear behavior can come from the crane's drive system. Some common examples are: acceleration limits, non-linear braking, friction, as well as nonlinear effects of the power/control electronics. Standard input shaping is designed for linear

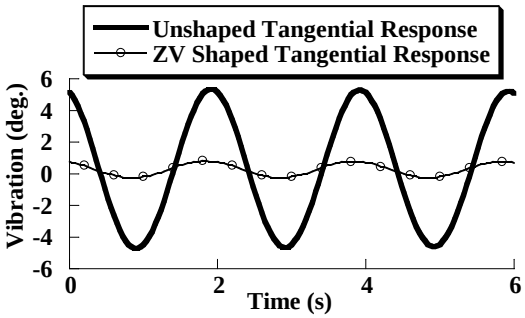
[1]Nonlinear payload dynamics can also occur in bridge cranes, although not for small swing angles



(a) Overhead View of Tower Crane.



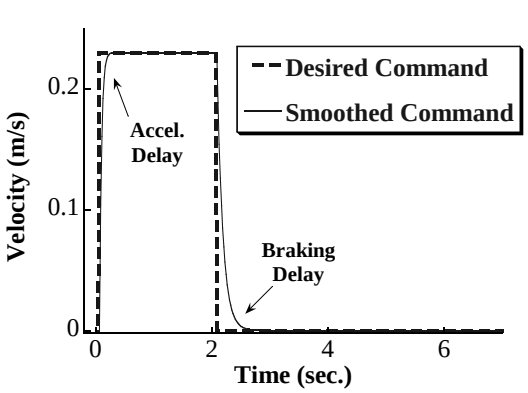
(b) Radial Vibration from 35^0 move.



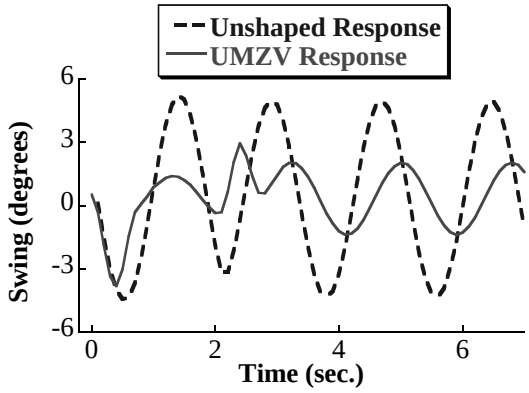
(c) Tangential Vibration from 35^0 move.

Figure 1.3: Transport Operation for a Tower Crane.

systems, these effects also degrade its performance. As an example consider the braking nonlinearity shown in figure 1.4(a). When the crane is commanded with a single velocity pulse, the actual trolley velocity cannot follow the command exactly. Instead, it accelerates to speed and decelerates to a stop at different rates, making the response nonlinear. Figure 1.4(b) shows experimental data taken from a miniature bridge crane. The crane has a height, width, and length of 1 meter, and is equipped with a camera to measure payload vibration. The payload swings due to the unshaped command as expected. When a shaped command is used the payload vibration is reduced, but not eliminated. Once again, the input shaper's performance has been degraded by the nonlinearities in the system. This motivates the questions: *Can an input shaper be designed to compensate for a braking nonlinearity? Can input shapers be designed for other types of drive nonlinearities?*



(a) Desired and Actual Commands with Braking Nonlinearity.



(b) Payload response to Unshaped and Shaped Commands.

Figure 1.4: Braking Nonlinearity Command and Response.

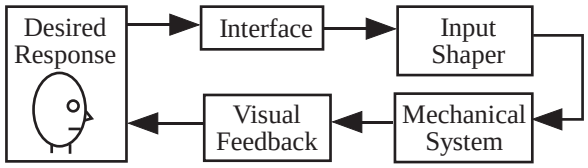


Figure 1.5: Operator-in-the-Loop Control.

The ubiquitous presence of cranes puts a high value on solutions to the problems mentioned above. The basic system under consideration is shown in Figure 1.5. The major elements that move the crane include: the human operator that generates the desired response, the interface that the operator uses to control the machine, and the input shaper that transforms the desired response into a signal that controls the motors and drives. The human operator then uses visual feedback to adjust the control input. As research progresses on this control architecture, two questions emerge: *How does the human operator respond to this system? How to teach design and evaluation of such systems?*

The first question focuses how crane operator performance is effected by input shaping. Simulations and experiments have shown that input shaping can reduce vibration. But in order to really understand its benefits in the field, it must be tested be human operators. Some of the important issues are: How well does an operator adapt to using input shaping? With input shaping enabled, will the crane operator be able to reduce task completion time and number of payload collisions? What are the benefits to using input shaping in remotely

operated crane systems.

The second question is directed toward engineering education. New technology paradigms demand new types of education. The challenges imposed by teaching material related to Figure 1.5 motivated the development of new advanced controls courses at Georgia Tech, GT Lorraine, and Tokyo Tech. The development of labs were an essential part of these courses to give the students hands-on experience and make the concepts seem more tangible.

1.3 Goals and Contributions

The goal of this thesis is to study and contribute solutions to the problems described previously:

Problem Current input shaping technology is not designed for nonlinear crane dynamics.

Problem Input shaping has not been well integrated into the engineering curriculum.

Problem The effects of input shaping on crane operator performance is not well established.

Problem Experimental crane setups are not available for testing nonlinear input shaping algorithms nor teaching input shaping in the classroom.

The thesis contributions directly address the problems listed above, namely:

Contribution New command shapers are developed to compensate for nonlinear dynamics. Two common sources of nonlinear crane dynamics are targeted: nonlinear dynamics in the drive system, and nonlinear dynamics in tower cranes. These new command shapers are designed to have the same strengths as standard input shapers. Both of these techniques can also be applied to other systems such as pick-and-place robots, circuit board assembly robots, boom cranes, as well as many more.

Contribution This thesis contributes new laboratory procedures and exercises designed to teach input shaping in the engineering curriculum. The thesis will also focus on how input shaping is taught and learned in the field.

Contribution Crane operator studies are performed to establish how input shaping effects operator performance in a wide variety of scenarios.

Contribution Three experimental setups were constructed and customized to the needs of each task.

Chapter 2 will review basic background material on input shaping and cranes that form the foundation for the remainder of the thesis. Chapter 3 will review the three experimental crane setups used throughout this thesis. These setups were used to test the proposed input shaping algorithms, as educational tools by students, and as testbeds for the crane operator studies. The accomplishments in input shaping education are discussed in chapter 4. Chapter 5 reviews the crane operator studies and discusses the conclusions that can be drawn from them. The problem of input shaping for tower cranes is developed over the course of the next two chapters. First, Chapter 6 will discuss the simpler, yet related, problem of input shaping a one-mass flyball system. Then, Chapter 7 will address input shaping for tower cranes, revisiting many of the ideas generated from the flyball analysis. A new approach for input shaping nonlinear drive dynamics is developed in Chapter 8.

CHAPTER II

BACKGROUND

2.1 Cranes and Crane Control

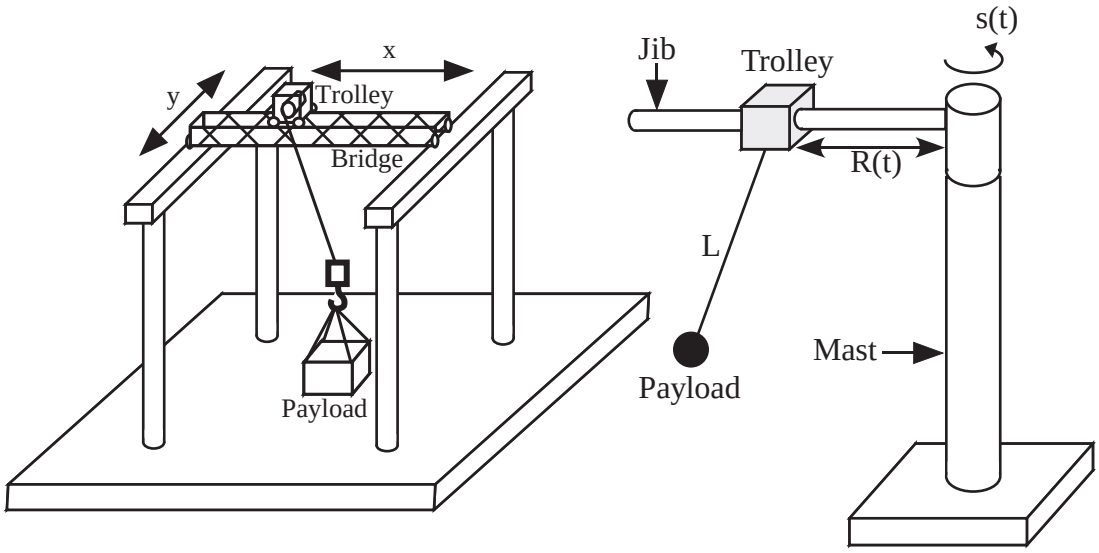
Two types of cranes are discussed throughout this thesis: bridge cranes and tower cranes. The two cranes are distinguished by their physical structure. A picture of a bridge crane was shown in Figure 1.1(b) and a sketch of one appears in Figure 2.1(a). In a bridge crane, the trolley moves across a beam or girder called the bridge. In the figure, this motion is labeled x. The bridge, in turn, can move on the surrounding support structure in a direction perpendicular to the trolley motion. In the figure, this motion is labeled y. The trolley supports the suspension cable attached to the payload. The trolley will often contain a hoisting motor that can raise and lower the payload.

A sketch of a tower crane appears in Figure 2.1(b) and a picture was shown earlier in Figure 1.1(a). In a tower crane, the trolley moves on a beam called the jib. The jib is attached to the top of a central beam or mast. The jib rotates around the mast.

Both bridge and tower cranes have three degrees of freedom, but their motions are inherently different. They are both capable of hoising the payload up and down. However, the bridge crane moves the trolley using two, linear, orthogonal motions. In contrast, the tower crane moves the trolley using a radial and rotational motion.

Bridge and tower cranes are used for different applications. Bridge cranes are typically found in warehouses, steel mills, assembly lines and shipyards. Often, they are built into the ceiling of a building as shown in Figure 1.1(b). Tower cranes are most commonly used in construction sites and are often assembled on site. Their small footprint on the ground and large workspace make them ideal for these types of applications.

Due to their different physical structures, the dynamics of bridge and tower cranes are different. The bridge crane dynamics are usually derived by decoupling the x and y motions shown in Figure 2.1(a). The payload response to each motion is derived separately using



(a) Bridge Crane Sketch.

(b) Tower Crane Sketch.

Figure 2.1: Bridge and Tower Crane Sketches.

the planar crane model shown in Figure 2.2. For example, the planar crane model can represent a single motion in the x-direction. The trolley is assumed to have velocity v_t and the suspension length is L. The angle between the suspension cable and vertical is θ. The equations of motion for this system are easy to derive and are stated without proof:

$$\ddot{\theta} + \frac{g}{L} = -\frac{1}{L}\dot{v}_t \tag{2.1}$$

Notice that the equations are linear and independent of the payload mass, similar to a pendulum. These equations are only valid for sufficiently small deflection angles. This

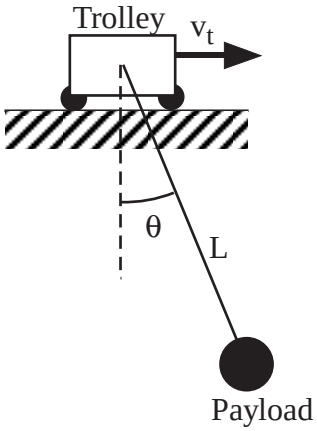


Figure 2.2: Planar Crane Model.

equation also assumes no damping in the payload swing. An identical equation can be found for motion in the y-direction. The payload response to a combination of x and y motions is found by adding the responses to each motion separately. This is a valid approximation for small deflection angles and trolley velocities.

The dynamics of tower cranes are inherently more complicated due to their rotational nature. Centripetal and Coriolis accelerations create nonlinear, coupled equations of motion. A derivation of these equations appear in Chapter 7.

As indicated by the above discussion, the payload response has very little damping. As a result, once the payload begins to oscillate it takes a long time for the oscillations to decay. This is not only time consuming and inefficient, but also poses a safety hazard. As a result, many control strategies have been developed for suppressing these vibrations.

2.1.1 Crane Control Methods

Crane control is a well studied problem. Several papers have been written on the subject and a thorough review of the literature appears in [1]. The short description below highlights only a few of the major contributions in this field.

Crane control methods can be loosely divided into three categories: command shaping, optimal trajectories and feedback control. The first method, command shaping, is the main focus of this thesis. A thorough review of command shaping for cranes is given in the next four sections of this chapter.

The second approach is to calculate the optimal crane trajectory ahead of time. The problem is usually formulated as a optimization problem that minimizes the time to reach the desired position. The start and end states of the system are constrained to move the crane to the desired position without vibration. This problem can also be viewed as a boundary value problem. Auernig and Troger [5] took this approach to find optimal trajectories for overhead cranes in shipyards. The problem was solved in closed form using Pontryagin's maximum principle. Golafshani and Aplevich [23] used a similar approach to find optimal trajectories for tower cranes. Their solution was obtained by discretizing the equations and using sequential quadratic programming. Similar techniques can be used for

command shaping, and will be discussed later.

The third approach is to use feedback control. Sensors feedback crane measurements (eg. deflection, position, etc...) to a controller that then generates the torque or speed command. Moustafa and Ebeid [63] used state feedback and a linearized state space model in their control system. The system was shown to be controllable and the closed loop poles were chosen to create fast, low-residual vibration response. Al-Garni, et. al. [3] also used state feedback, but the optimal control gains were chosen with numerical optimization. Butler, et. al. used an adaptive model reference controller [14] to tune the controller gains given a periodic input. A passive based control scheme was developed by Alli and Singh [4] and the controller parameters were selected via optimization. A wave-based approach was taken by O'Conner in [66]. The deflection waves of the payload are recorded while the trolley ramps up to speed, then played back through the controller to return the system to rest.

All of these techniques have advantages and disadvantages. The optimal trajectory strategy will find fast and effective commands. However, these techniques require that the desired move distance be known ahead of time and extra constraints must be added to make sure the trajectory is feasible. In addition, calculating new optimal trajectories can be computationally expensive and usually needs to be done off-line. The feedback control approach is robust and also includes disturbance rejection. However, sensors must be employed to measure the payload deflection and other system states. These sensors can be expensive and impractical. Feedback control can also sometimes cause unexpected motions that make it difficult for the operator to drive the crane. All of these problems can be resolved by using input shaping, as discussed in the next section.

2.2 Input Shaping Theory

Input shaping plays a central role in this thesis. This section will review how input shaping works and what makes it a good solution for crane vibration.

When an input shaper is used, it modifies the desired velocity command before it is fed to the crane drives. A typical implementation of a input shaper on a crane system is

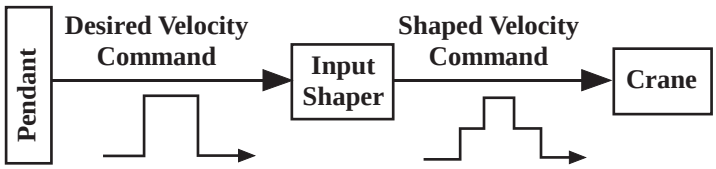


Figure 2.3: Block Diagram of an Input Shaped Crane System.

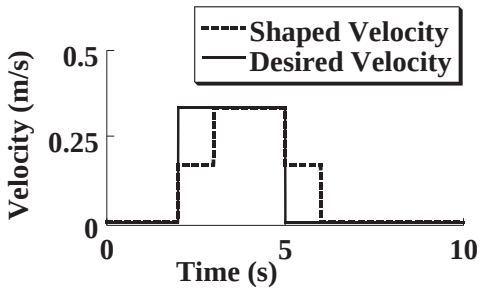


Figure 2.4: Desired and Shaped Velocity Commands.

shown in Figure 2.3. Figure 2.4 shows the original desired command and the new velocity command created by the input shaper. The desired command to achieve the point-point motion is a velocity pulse. In contrast, the shaped velocity command stays at half-speed for a short time before reaching the final full-on velocity. It is this modification, which the input shaper imparts on the velocity command, that causes the payload to move without swinging. The step changes in velocity are timed to cancel out the payload swing.

The previous example showed that input shaping works by modifying the desired velocity command. It achieves this by applying a convolution filter to the command. The input/output formula for a convolution filter is:

$$Y(t) = G_{IS}(t) * X(t) \tag{2.2}$$

Where X(t) is the input, G_{IS} is the input shaping filter sequence, $Y(t)$ is the output, and the $*$ is shorthand for the convolution integral. All input shapers use a sequence of impulses, with different times and amplitudes, for G_{IS}. A generalized input shaper is shown in Figure 2.5.

We will now examine how input shapers are derived and why they move systems with zero vibration. For the moment, consider what would happen if the input shaper itself were

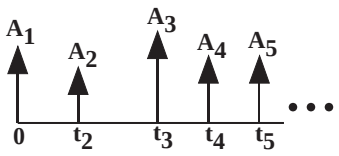


Figure 2.5: Generalized Input Shaper.

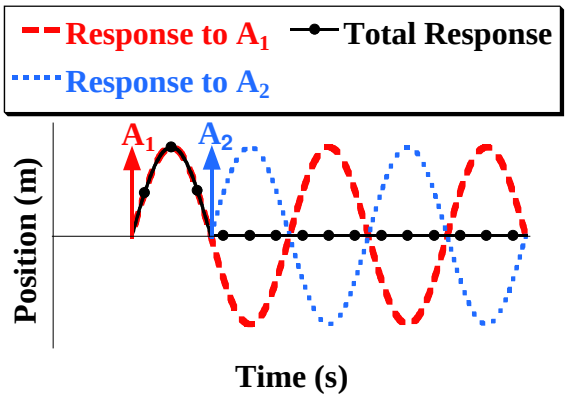


Figure 2.6: Superposition of Impulse Responses.

used to command the crane system. Each impulse would excite some amount of vibration. This process is shown in Figure 2.6 for a two-impulse sequence. The dotted lines show the vibration produced from each impulse. However, because the system is linear we can use superposition to find the total response to all the impulses. If the impulses are chosen correctly, the vibration from each impulse will cancel out. This is shown as the solid black line in Figure 2.6. One of the first solutions to this problem was described by O.J.M Smith in the 1950's [94].

The same procedure can be applied to the generalized input shaper shown in Figure 2.5. The goal is to find the impulse times and amplitudes such that the total response has zero vibration. The impulse times and amplitudes can be solved numerically or, in many cases, in closed form. In either case, the impulse times and amplitudes will be a function of the system natural frequency and damping ratio. An extensive review of these solution techniques is given in [80]. Some basic analytical techniques are given in [81, 88] and a discrete optimization approach is discussed in [76].

If an input shaper is used to drive the system, it will cause zero vibration. However, as Figure 2.3 showed, an input shaper alone is never used to drive the system. It is used

as a convolution filter. So the real question is, why does the output of the convolution filter move the system without vibrating? One way to show this is by looking at Fourier transform of (2.2):

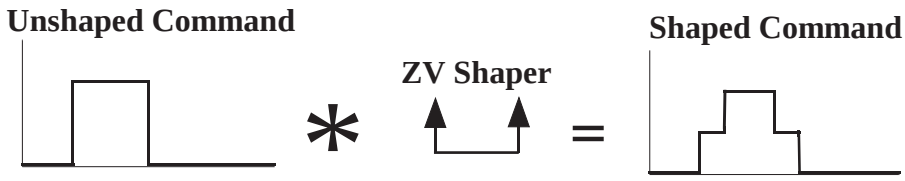
$$Y(\omega) = G_{IS}(\omega)X(\omega) \tag{2.3}$$

Notice that the convolution operator has been transformed into multiplication. Recall that the input shaper was chosen such that it excites zero vibration at the system's natural frequency or $G_{IS}(\omega_n) = 0$. Therefore, it follows from (2.3) that $Y(\omega_n) = 0$. In other words, the shaped output, $Y(t)$, always has zero frequency content at the system's natural frequency regardless of the input $X(t)$. It has been shown in [7] that if the input to an undamped system has zero frequency content at the system's natural frequency, then the system will not vibrate. This completes the proof. Note that if $X(\omega)$ has poles at the zeros of $G_{IS}(\omega)$, then pole-zero cancelation will occur. Under these conditions, the resulting $Y(\omega)$ may have non-zero frequency at the system's natural frequency causing the system to vibrate.

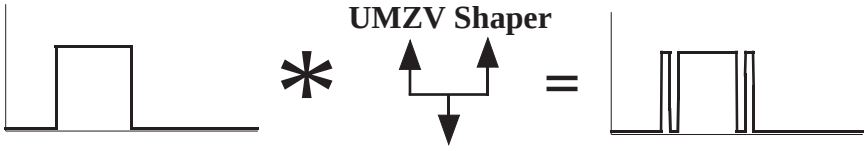
2.2.1 Input Shaping Properties

Input shapers have several characteristics that make them easy, effective solutions for eliminating vibration in cranes. For instance, any desired velocity command can be used and input shaping will always eliminate the vibration. In addition, different types of input shapers can be used in the convolution filter. Figure 2.7 shows three different shaper types, along with the resulting shaped pulse command. Each has different properties that make it favorable for different scenarios.

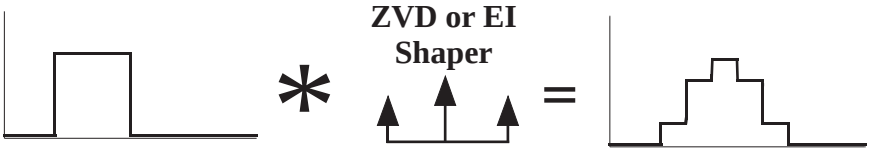
- Figure 2.7(a) shows the simplest type of input shaper, a Zero Vibration (ZV) input shaper.
- Figure 2.7(b) shows a Unity Magnitude Zero Vibration (UMZV) input shaper [88]. This shaper is designed to be fast and also is ideal for on/off control systems.
- Figure 2.7(c) shows a Zero Vibration Derivative (ZVD) [81]. This shaper is robust to modeling errors in the system's natural frequency.



(a) ZV Shaped Pulse.



(b) UMZV Shaped Pulse.



(c) ZVD or EI Shaped Pulse.

Figure 2.7: Various Shaped Pulse Commands.

Another shaper, not shown in the figure, is an Extra-Insensitive (EI) shaper [84, 85]. Its impulse sequence is essentially the same as a ZVD shaper, but with slightly different impulse amplitudes. This shaper is more robust to frequency modeling errors than the ZVD shaper and is formed by relaxing the zero-vibration constraint at the modeled frequency.

Notice that all of the shaped commands consist of step changes in velocity. It may seem counter-intuitive to use such commands since step changes in velocity tend to excite vibration. Many systems use smoothed commands to eliminate residual vibration. Eloundou and Singhose [18] compared a type of smoothed command, called an S-curve, to input shaped commands. Their conclusion is that smoothed commands can be tuned to have the same vibration-reduction properties of a input shaped command, but the result will always have a longer rise time than the shaped command. Therefore, there is never any benefit to using a smoothed command over a shaped command. Consequently, all of the shaped commands developed in this thesis for nonlinear systems are assumed to have step-changes in velocity.

One important property of input shapers is robustness to frequency modeling errors. This is important because the natural frequency of a system may not be known exactly. Furthermore, it is useful for a shaper to perform well if the system frequency changes. In a crane system the frequency can change by hoisting the payload or attaching a payload that adds another vibration mode. The robustness of a shaper can be measured using a sensitivity plot [81]. Sensitivity plots for the four aforementioned shapers are shown in Figure 2.8. A sensitivity plot shows the system residual vibration of a input shaped command for various system frequencies. Most frequently this is curve is derived theoretically using a closed form expression for the residual vibration of an input shaper on a linear system [81]. However, it can also be experimentally plotted for a real system [78]. The horizontal axis in Figure 2.8 shows the normalized frequency, which is defined as the system frequency divided by the model frequency. The vibration measured at a normalized frequency of 1 indicates the shaper's performance at the modeled frequency. The vertical axis shows the normalized vibration, which is defined as the residual vibration for the shaped command divided by the residual vibration for the unshaped command. For linear systems, the curves shown in the figure are the same regardless of the unshaped command[1].

The sensitivity of a input shaper describes how the shaper performs due to un-modeled frequency variations in the system. The width of each curve in Figure 2.8 is an indication of shaper sensitivity. For example, suppose a 95% reduction in vibration was desired. According to the figure, with an EI shaper, the system frequency can be varied by $\pm 20\%$ from the modeled frequency before exceeding the 5% normalized vibration level. In contrast, the ZV shaper can only be varied by $\pm 3\%$. These ranges can be used to define the insensitivity of a shaper [85]. Based on these results, the EI shaper is more robust to modeling errors, making it more desirable to use. However, recall from Figure 2.7 that the EI shaper has a longer duration than the ZV. The trade-off between robustness and shaper duration is a common design trade-off in input shaping. In general, the sensitivity of the four shapers can be ordered as follows from most to least: UMZV, ZV, ZVD, EI.

[1]There are some unshaped commands that are exceptions. For example, a sinusoidal command with a frequency equal to the natural frequency of the system.

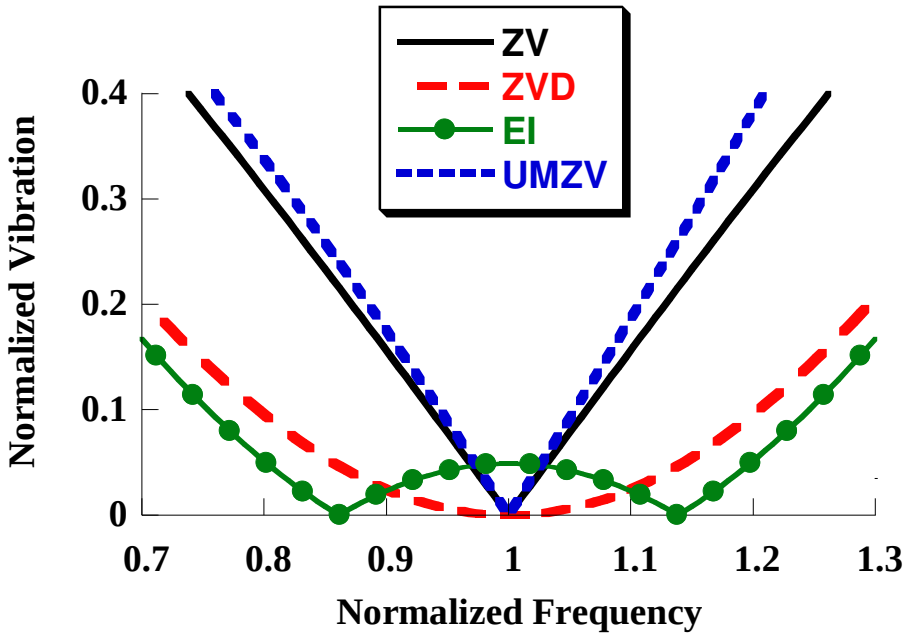


Figure 2.8: Sensitivity Curves for Various Shapers.

The sensitivity plot can be used to design shapers as well. Singer and Seering [78] created robust shapers by numerically constraining the sensitivity curve to be below a threshold at a series of sample points. Singhose, et. al. then developed a procedure for finding exact solutions of shapers that remain below a given threshold across a frequency range. These shapers were termed Specified-Insensitivity (SI) shapers.

Another important characteristic of input shapers is that the algorithm is easy to implement on real-time systems. An input shaper relies entirely on the known system parameters, so no extra sensors are needed to provide feedback information. The convolution algorithm is easy to program, and does not require large amounts of processing power. In addition, new velocity setpoints can be fed to the input shaper on-the-fly, making it ideal for human-operated systems like cranes.

2.3 Input Shaping and Cranes

Input shaping has had a tremendous impact on cranes. Due to its oscillatory nature, a crane's response can be vastly improved using input shaping. This section reviews the some of the previous research on applying input shaping to cranes.

2.3.1 Background: Basic Input Shaping for Cranes

One of the first papers to suggest using input shaping for cranes was published by Starr in 1985 [98][2]. In this paper he suggests using a very simple method for canceling vibration that is akin to a ZV shaped pulse command. Since then, numerous papers have implemented basic shaping techniques on cranes and demonstrated its ability to suppress payload oscillation. Robinett, et. al. [77] and Noakes, et. al. [65] implement a basic shaping scheme on gantry and jib cranes at Sandia National Labs and Oakridge National Labs. Input shaping has also been implemented on the HiBay crane at Georgia Tech [86, 95].

Several studies have examined the performance of input shaping on cranes. Singhose et. al. [87, 90] compared the performance of input shapers with time optimal commands. It was shown that compared to the time optimal trajectory, input shaping reduced both the residual vibration after the move, and the transient deflection during the move. It was also shown that input shaping was effective for hoisting motions, even when the hoisted distance was a large percentage of the suspension length. Similarly, Park, et. al. [69] showed that input shaping was effective when the suspension length changed during hoisted transport operations of container cranes. At the Savannah River Technology Center, Singer et. al. [78] developed and implemented a new Fixed Duration input shaper that suppressed residual vibration for a large range of suspension lengths. Experimental results confirmed the robustness of the new shaper to suspension length.

2.3.2 Input Shaping Variations for Cranes

Several crane-related input shaping techniques have been developed that go beyond the standard implementations cited above. For example, the payload and its rigging often create a second vibration mode causing the crane to exhibit double-pendulum dynamics. Singhose, et. al. [89] derived multi-mode shapers that targeted the frequency of each mode. Kennison et. al. [32] and Kim et. al. [38] designed Specified Insensitivity (SI) shapers that suppressed the vibration across a range of frequencies that included the frequencies of the double pendulum modes.

[2]However, Smith [93] published a figure in his book that shows input-shaped crane structure in 1958.

A number of papers have taken a different, albeit related, approach to command shaping. Rather then convolve the desired command with an impulse sequence, a notch filter is designed that has zeros at the system poles [2, 55, 56]. As mentioned earlier, an input shaper achieves the same goal [7]. Kress et. al. [46] applied this technique to an overhead crane. It has also applied to rotary boom cranes [2, 55, 56]. However, there are several downfalls to this technique. The notch filter often adds additional smoothing that results in a slower command compared to standard input shaping. In addition, the rotary boom crane studies used a complex (and impractical) control system to account for the nonlinear dynamics of these cranes.

Some studies have also explored combining input shaping with feedback control in cranes. The advantage is that the feedback control can compensate for nonlinear effects as well as disturbance. Hong, et. al. [28] used a sliding mode controller in conjunction with input shaping to eliminate the nonlinear effects of hoisting. Sorensen, et. al. [95–97] used input shaping, PD control, and model of the system for disturbance rejection and accurate position control of a gantry crane.

2.4 Input Shaping for Nonlinear Systems

A primary focus of this thesis is developing input shapers for nonlinear systems. Because input shaping is developed for linear systems, its performance is degraded when used on nonlinear systems. Several previous studies have examined input shaping for nonlinear systems. In the descriptions that follow these studies are categorized, reviewed, and evaluated for their strengths and weaknesses. Some of these techniques form a basis for the shaping algorithms developed in this thesis.

2.4.1 Robust Shaping

One of the simplest shaping techniques for nonlinear systems is to employ robust input shapers. The motivation is that shapers which are designed to be robust to variations in frequency are also robust to nonlinear dynamics. This strategy works particularly well when the non-linearity involves a time or position dependent natural frequency.

Crane, et. al. [15] showed the effectiveness of EI and ZVD shapers on a serial, 2-link

robotic arm with position dependant frequencies. Similarly Kozak, et. al. [45] showed the effectiveness of SI shapers on a parallel manipulator with configuration dependant frequencies. Stergiopoulos, et. al. [99] showed the effectiveness of a ZVD shaper on a crane with a nonlinear pneumatic drive. Singer and Seering [78] developed a robust shaper using frequency sampling, similar to an SI shaper, and showed its effectiveness on a space shuttle robot arm with a configuration-dependent frequency.

The major strength of this approach is its simplicity. The shapers used are no different that those for linear systems, so no new shaping solutions need to be derived. The control architecture is also the same as a standard input-shaped linear system. The disadvantage is performance. Because these shapers are designed using linear system theory, the amount of residual vibration will depend on the severity of the nonlinearity. For large nonlinearities there will be significant residual vibration. In addition, there is no way to know whether a robust shaper is good choice for a nonlinear system without running several simulations or experiments.

2.4.2 Adaptive Shaping

An adaptive shaper automatically tunes the impulse times and amplitudes based on measurements of the system output. Park, et. al. [70] applied this technique to a basic 2^{nd} order system with time-varying coefficients. The idea was to measure the amplitude and phase characteristics of the output, and adaptively change the shaper until these measurements matched a standard ZV shaper. This idea is very similar to the adaptive technique developed in Chapter 8. However, their solution is limited to ZV type commands and their algorithm is different from the one developed in this thesis.

Many more adaptive techniques have been applied to both linear and nonlinear systems. Because this section deals strictly with nonlinear techniques, a more thorough discussion of adaptive technique appears in Section 2.5.

2.4.3 Linearization

Linearization methods use a linearized model of the system to estimate its natural frequencies and damping ratios in various configurations. These methods can be divided into two

approaches. One approach is to derive the frequencies/damping-ratios of the linearized model for each point in the trajectory and create a configuration-dependent shaper that changes as the system moves through each point. This technique can also be considered "adaptive" in that the shaper adaptively adjusts the parameters based on the current configuration. The second approach is to calculate the range of frequencies/damping-ratios encountered in the pre-known trajectory from the linearized model. Then, the shaper is designed for the "mean-value" of these parameters, or a robust shaper is designed to suppress the entire range of parameters.

Kozak, et. al. developed and applied a linearization strategy to a 2DOF, planar, parallel manipulator [43–45]. Both "mean-value" and robust shapers were designed based on this linearization process. Beazel and Meckl [6] created a configuration-dependent shaped command by dividing the command into segments and shaping each segment with the locally determined frequencies. Magee and Book developed a configuration-dependent shaping technique for a 2-link, flexible manipulator [60–62]. Rather than linearize the plant analytically, the robot was moved through the trajectory ahead of time and the natural frequencies and damping ratios were determined from the FFT of the response. Stergiopoulos and Tzes [100] used a linearization approach to develop shapers for a pendulum with large deflection angles.

Linearization techniques work well because they account for the effect of the nonlinear dynamics. Of the two shaping strategies, the configuration-varying shaper approach has the most shortcomings. The exact command must be known ahead of time and it is often unclear when the nonlinear system transitions from one linearized model to the next. Magee and Book showed that there are also implementation issues regarding convolving the desired input with a time/configuration-varying shaper. The "mean-value" or robust approach are easy to implement, well defined, and have been shown to yield good results. This thesis will implement such a strategy on rotating systems: the 1-mass flyball system, and the tower crane.

2.4.4 Combining Input Shaping with Other Control Elements

Some studies have examined combining a input shaper with other control elements. One strategy is to use a feedback controller to compensate for the nonlinear dynamics, either by linearizing the plant or treating the nonlinear dynamics as a disturbance. The input shaper eliminates the vibration from the linear portion of the plant.

Rhim et. al. used a multirate repetitive learning controller to eliminate the nonlinear effects of friction in a simple 1 DOF robot with a flexible appendage [75]. Then, an OATF filter, which is a special type of input shaper, was used to eliminate the vibration in the flexible appendage. Khorrami et. al. [36,37] used a nonlinear controller to linearize a multi-link flexible manipulator to improve the performance of a adaptive input shaper. Hillsley and Yurkovich [27] used a combined shaper and feedback control strategy for moving a two-link flexible arm. An input shaper was used for the gross motion, then a feedback control was engaged to eliminate any residual vibration at the end of the move due to the nonlinearities. This technique has also been applied to cranes. Hong et. al. [28] and Sorensen et. al. [95] also combined input shaping with feedback control for crane systems, as mentioned earlier. Kenison and Singhose [31] showed that a input shaper could be combined with PD control for a simple force-driven mass. Although this technique was applied to a linear system it could modified to work for simple nonlinear systems.

Another strategy is to employ inverse and forward kinematic converters to linearize system dynamics in the control loop. This strategy is particularly well suited to systems with a joint or kinematic nonlinearity, such as a robot arm. The idea is to take the desired command in joint space and convert it into the desired path in cartesian space. Because motions in cartesian space are linear, a standard input shaper will eliminate residual vibration. Then, the shaped command is converted back into joint space. This strategy was used by Singer [80] for input shaping a two-link, manipulator. It is similar to the techniques from the preceding paragraph in that the control elements are essentially linearizing the system by converting the command into cartesian space.

These techniques are very effective due to the robustness of the feedback control. However, a big disadvantage is that they require extra sensors and a more complex control architecture. Therefore, combined shaper and feedback control strategies will not be addressed in this thesis. In addition, the added control elements, particularly for the inverse/forward kinematic case, can sometimes cause undesired motion in the response.

2.4.5 Boundary Value and Numerical Optimization Techniques

The problem of input shaping nonlinear systems can be treated as a boundary value problem (BVP) which can be solved using optimization. The start and end points of the system, along with the zero-vibration end-point constraint, form the initial and final conditions of the BVP. The system dynamics are also assumed to be known in closed form. The problem then becomes a classical BVP: find the input which moves the system from the initial to the final states given the dynamic equations of motion. The literature has developed two possible approaches to solve this problem. The "inverse dynamics" approach is to form a polynomial-guess for what the desired trajectory should look like, substitute into the equations of motion, and derive the torque command. The second approach is to use optimization methods to solve the problem.

Kinceler and Meckl [40] used both an "inverse dynamics" approach and a optimization approach to control a two-link robot. Kinceler and Meckl [39] then performed a second study that developed a improved "inverse dynamics" approach by dividing the trajectory into segments, and used a feedback controller to eliminate errors. Smith et. al. [92] used a piecewise optimization approach to find the times and amplitudes of a ZV shaper for a system with a linearly changing natural frequency. Similarly, Kozak et. al. [45] used large scale optimization to find the shaper parameters for a two-link, planar parallel manipulator. Lim and How [58, 59] used convex optimization to find commands that could move a two-link, flexible manipulator that had acceleration and velocity limits (saturation nonlinearity). Eloundou and Singhose [19] developed a numerical optimization approach to find shaper parameters for PID controlled systems with saturation by minimizing the residual vibration.

Boundary value methods succeed in developing commands that eliminate vibration,

and can be applied to broad range of nonlinear systems. But these methods frequently require optimization that can be computationally intensive. Another drawback is of these methods is that most of them target finding zero-vibration commands and require additional constraints to form robust commands.

2.4.6 Closed-Form Solutions

Exact or closed-form solutions for input shaped commands can be found if a model of the nonlinearity is known ahead of time. This strategy is carried out by evaluating the effect of the nonlinearity on a linearly shaped command. The mathematical model of the system is used to find shaper values that compensate for these nonlinear effects. The result is a closed-form solution for the shaper parameters, usually given as a function of the system parameters.

Singh [82] used this approach to find several different types of shaped commands for jerk-limited systems. This approach can be used to find zero vibration and robust commands for a PD controlled mass with Coulomb friction [26, 49, 51, 52, 54]. A closed-form solution was also found for systems with backlash [50].

Closed form solutions can be very effective, particularly considering that the shaper parameters can be solved directly form a closed-form expression. One disadvantage is that closed-form solutions often make several assumptions about the nonlinear dynamics. In addition, each solution only applies to systems with the modeled nonlinearity and the approach to solving for the shaper parameters is different for each system. In some instances there may be no closed-form solution.

2.5 Adaptive Input Shaping

One of the new nonlinear shaping techniques discussed in Chapter 8 uses a simple adaptive technique to arrive at a solution. Several previous studies have designed input shapers using adaptive techniques. All adaptive techniques share the same basic procedure: move the system, measure the response, use the measurement to design the input shaper and/or controller parameters, and repeat. A review of the literature has shown that these techniques fall into three basic categories:

- Move the system and identify the natural frequency and damping ratio. Then use these parameters to calculate the input shaper impulse times and amplitudes (Rhim and Book classify this as a *indirect* technique [74]). Tzes, et. al. [35, 102] use a frequency domain technique to identify the system in real-time. Similarly, Kojima et. al. [41,42] measured bending moment of a space structure to determine the natural frequencies in real time. These were then used to create an adaptive input shaped command with a specified deflection limit. Bodson used a Recursive Least Squares technique in the time domain to identify the parameters both offline [10] and in real-time [11].

- Move the system and choose shaper parameters that will make the actual system response match the ideal shaped response (Rhim and Book classify this as a *direct* technique [74]). Rhim and Book use [73, 74] a Recursive Least Squares technique to match the time response of the actual system and ideal shaped system. Park and Chang [71] measure the amplitude and phase of the response and try to match it to the amplitude and phase of an input shaped response. Cutforth and Pao [16] use a combination of Rhim and Park's techniques.

- Adapt the system, rather than the shaper. In [17] parameters of a 1^{st}-order controller are adjusted so the closed loop poles match the input shaper zeros. Rhim, et. al. [75] used a repetitive learning controller to eliminate the nonlinear effects (mostly friction) in the rigid body mode while the command shaper eliminated the vibration of the flexible mode.

All of these techniques have shortcomings. For instance, most of these techniques need a measurement of the end-point (payload) vibration, which is often not available for cranes. Furthermore, input shaping doesn't normally need end-point sensors, so using one would eliminate one of its main strengths. In the adaptive algorithm developed in Chapter 8 only the trolley velocity needs to be measured, not the swinging payload. The trolley velocity is much easier to measure and in some cases is already measured by the drive. For flexible machines, in general, it is usually easier to measure the drive output rather than the vibration of the flexible component.

Another shortcoming of the aforementioned studies is that they all develop a Zero-Vibration type solution. This is not practical if a robust shaper will be used to cancel vibration across a range of payload heights [79] or if a multimode shaper is used for payloads with multiple modes (eg. double pendulum effects) [32]. Finally, all of these techniques, with the exception of [71, 75], assume the plant is linear, which is not the case for many of the systems considered in this thesis.

CHAPTER III

EXPERIMENTAL CRANE DESIGNS

3.1 Overview:Crane Projects

Three cranes were developed as experimental testbeds for this thesis. These machines were used to get experimental results for the new input shaper algorithms developed in this thesis. Additionally, these cranes were used by students all over the world to teach input shaping and explore crane operator behavior. A significant amount of design work was performed to create these testbeds. This chapter focuses on these three cranes and describes their capabilities and how they were developed:.

A summary of each crane project appears below in chronological order:

HiBay Crane [Completed 2003] An existing 10-ton Bridge crane in the HiBay of the Georgia Tech MARC Building was modified. Siemens motors and drives were installed and a Siemens PLO was programmed to control the motion of the crane. The crane has a workspace of 100 ft. x 30 ft. with a 20 ft. hoist.

Bridge Crane [Completed 2004] A portable bridge crane was created with dimensions 1 m x 1 m x 1 m. The system is controlled using Siemens motors, drives, and PLO. A digital camera was added to records the payload swing. This system was shipped to GT Lorraine, in France where it was used as an instructional and research tool.

Tower Crane [Completed 2005] A portable Tower crane was created that is 2 m tall with a 1 m jib. The system is controlled with Siemens motors, drives, and PLO and uses a digital camera to record the payload swing. The system also has a GUI interface and is capable of being controlled over the internet. The tower crane was shipped to Tokyo Tech. in Japan where it was used as an instructional and research tool.

Each crane had its own design challenges that needed to be solved. Figure 3.1 shows the major design challenges of each project. In the HiBay crane project, the main challenge

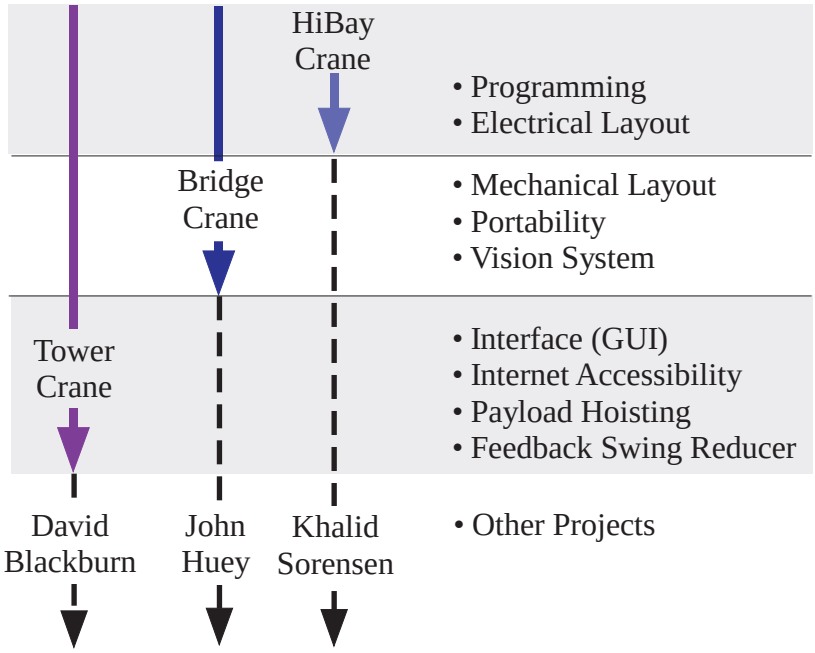


Figure 3.1: Progression of Crane Projects

was designing the electrical system and programming the input shaping algorithm. In the second project, the bridge crane, the primary challenge was designing the entire mechanical system from the ground up and making it portable. In addition, a digital camera had to be configured to measure the payload swing. In the third project, the tower crane, the additional challenges were designing the graphical user interface (GUI) and making the system capable of being controlled over the internet. Other challenges included the design of the hoisting system and implementation of a feedback control swing reducer.

The solid arrows in Figure 3.1 represented my contribution to each crane project. However these crane projects also laid the foundation for projects by other graduate students, shown as dotted lines in the figure. The HiBay crane project was continued by Khalid Sorensen, John Hued continued research with the bridge crane, and David Blackburn used the tower crane in his research. Additional work continues to be done by other graduate and undergraduate researchers as well.

Each crane was increasingly more complex, adding to the functionality of its predecessor. The knowledge learned from each project carried over to the next. This is also reflected in Figure 3.1. For example, knowledge of how to configure and program the electrical system

carried over from the HiBay project to the bridge and tower crane designs. The vision system on the tower crane uses the same algorithm as the bridge crane. Experience with making the bridge system portable, influenced how the tower crane was designed to be portable. This made the design process for each project more efficient.

In the sections that follow, each of the three crane projects will be described in more detail. Each section highlights the major design challenges that were overcome. Design components that are repeated from previous crane projects will not be described a second time. Figure 3.1 can serve as a guide for what design components carried over from one project to the next. For example, the input shaping program for the HiBay crane will carry over to the other projects and, therefore is not repeated. The chapter ends with some lessons learned in the course of designing these systems.

3.2 HiBay Crane

The HiBay crane is a 10-ton bridge crane in the HiBay of the MARC building at Georgia Tech. A sketch of the crane is shown in Figure 3.3 while a picture of the crane is given in Figure 3.2.

Prior to this thesis, the crane functioned like any other standard bridge crane. An operator would push the directional buttons on the pendant and the crane would move in the desired direction. Pushing Left or Right would move the trolley across the bridge beam, shown in Figure 3.3. Pushing Forward or Backward moves the entire beam orthogonal to the trolley motion. The crane was also capable of hoisting up and down.

The goal of the HiBay project was to install input shaping on the crane. The operator's commands would be automatically convolved with an input shaper and used to drive the crane. To achieve this goal tasks needed to be completed: 1) an electrical system needed to be designed and installed that used new motors, drives, and a PLO 2) The PLO needed to be programmed with input shaping.

3.2.1 Electrical Hardware Description

Figure 3.4 gives an overview of the hardware configuration used for the HiBay Crane. When the operator pushes the pendant buttons it sends digital commands to the PLO. The PLO

Figure 3.2: HiBay Crane Picture.

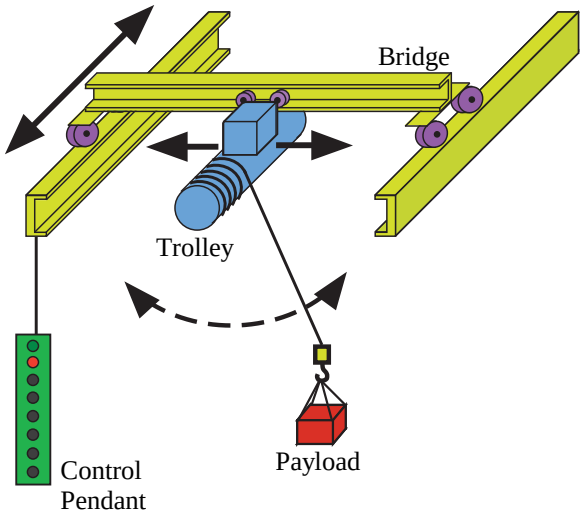


Figure 3.3: HiBay Crane Sketch.

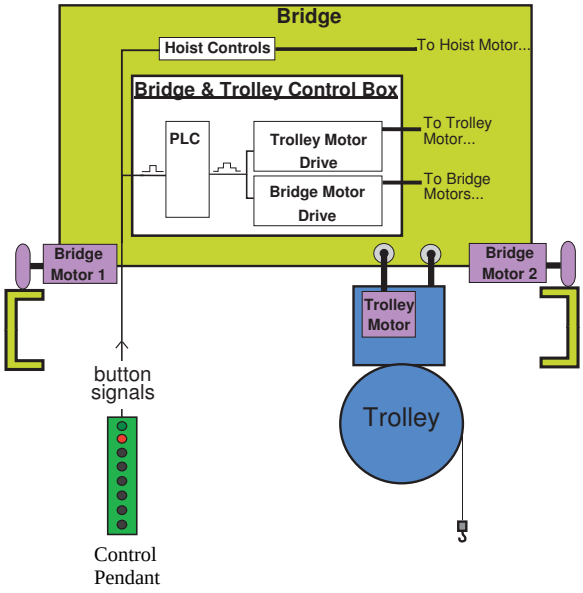


Figure 3.4: HiBay Crane Electrical Hardware Overview

convolves these commands with an input shaper. The shaped command is then sent to the drives as an analog voltage signal. This voltage signal serves as the desired velocity command for each axis. The drives power the motors and try to track this desired velocity command using an internal feedback loop.

The existing motors on the HiBay crane were incompatible with the hardware layout shown in Figure 3.4 and needed to be replaced. The crane had asynchronous, AC motors controlled by relays. When the operator pushed one of the buttons, a relay would trip, thereby sending power to the motors. The problem with this setup was that asynchronous AC motors are not designed to track a complex velocity trajectory. Inside these types of motors the power signal generates a rotating magnetic field in a stationary wire-coil called the stator. However, for this field to create torque it must spin at a different rate than the shaft or rotor. This design causes a lag and steady state offset between the rotor speed and the desired speed. As a result of these properties, asynchronous motors typically have poor trajectory tracking and sluggish response (low bandwidth). In addition these motors are usually limited to turning at only 1 or 2 speeds due to the physical construction of the stator coils. These disadvantages would severely limit the scope of input shaping experiments that could be performed.

The solution was to install inverter-duty, asynchronous motors and AC-AC inverter drives. The inverter drives generate a high power, pulse width modulated (PWM) command that provides fast response. The inverter-duty motors are specially designed to handle these high-power, rapidly changing commands. The drives monitor the output current and voltage to the motor and use an internal model to estimate the motor speed. This is used to close a proportional-plus-integral (PI) speed control loop which provides relatively high bandwidth response and good trajectory tracking. Furthermore, the PLO can easily send commands to the drive through an analog input. Replacing the motors and drives to facilitate input shaping has been previously mentioned in the literature [79].

3.2.2 PLO programming

The input shaping algorithm is performed by the PLO. The algorithm starts by creating a buffer to store the velocity command - a vector variable of a finite length. A graphical representation of such a buffer at the first time step is shown in the upper right-hand corner of Figure 3.5. This buffer is used to store the command values for each time step. So, in this example, the current velocity command is shown in slot 1, the velocity command for the next time step is stored in slot 2, etc... The upper left-hand portion of the figure shows the unshaped initial command in the digital domain. This baseline command comes from the operator pendant buttons.

In order to fill the buffer with the input-shaped command, the algorithm acquires the baseline command each time through the control loop. Then, the algorithm multiplies the baseline command at that instant by the amplitude of the first impulse in the input shaper. This value is added to the current time location in the shaped-command buffer. The amplitude of the second impulse is then multiplied by the baseline command. However, this value is not sent directly out to the control loop. Rather, it is added to the future buffer slot that corresponds to the time location of the impulse. Although the figure uses two-impulse input shaper as an example, the process can be repeated for input shapers with more impulses.

This real-time process will build up the shaped command as demonstrated in the bottom

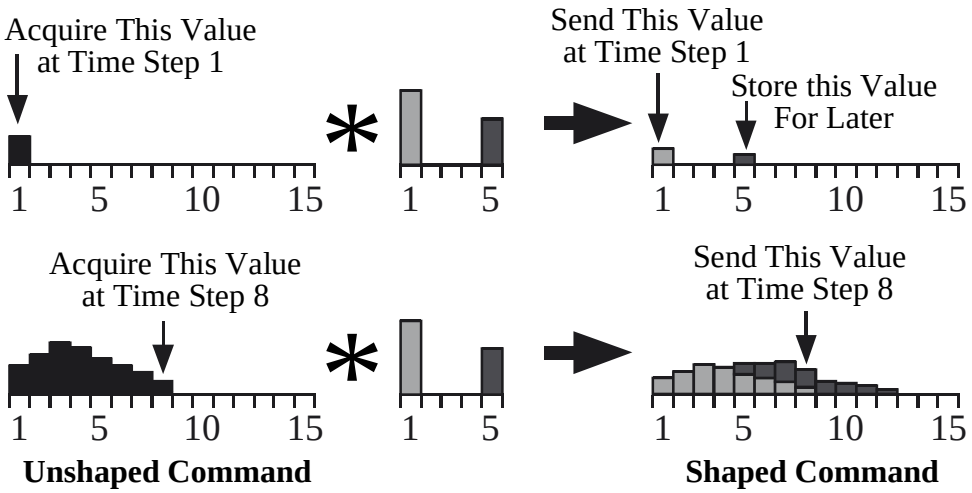


Figure 3.5: Real-Time input shaping.

of Figure 3.5. The bottom-right portion of the figure shows the state of the buffer at the eighth time step. To avoid having the index exceed the size of the buffer, a circular buffer is used where the index goes back to the beginning when it reaches the end of the buffer.

The input shaping algorithm described above was programmed into the PLO using the structure shown in Figure 3.6. The program consists of two loops; one running asynchronously shown on the left of the figure, and the other running with a cycle time of 20 ms shown on the right of the figure. The asynchronous loop acquires the button command from the pendant and performs the input shaping algorithm. Recall from Figure 3.5 that the input shaping algorithm fills a setpoint buffer. This setpoint buffer is labeled as *Data Block* and appears in the middle of the figure. Each slot of the *data block* represents the velocity command every 20 ms. The i is the time index value and the #### is the commanded velocity at the indexed time. The final step of the asynchronous loop is to write the input-shaped velocity commands to the corresponding time slots in the *Data Block*.

The synchronous loop is responsible for sending the velocity command to the drives every 20 ms. It reads the velocity command for the current time, indexed by a variable i. Then, this value is sent to the drives as a analog current signal. The final step is to increment the counter i, or reset it to zero if the end of the buffer has been reached.

The program shown in Figure 3.6 shows the input shaping scheme for a single axis.

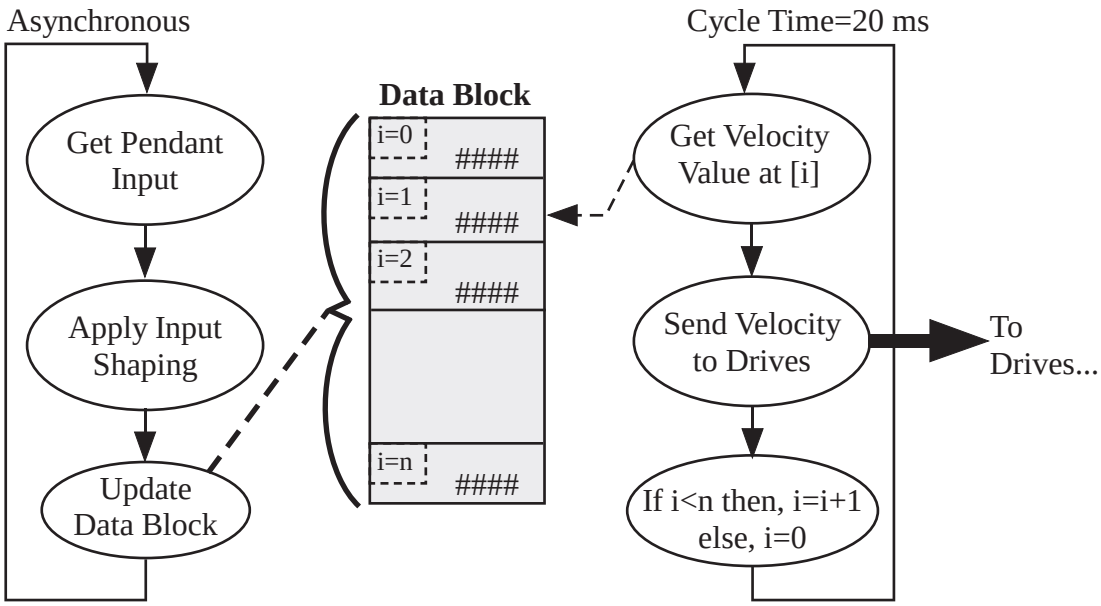
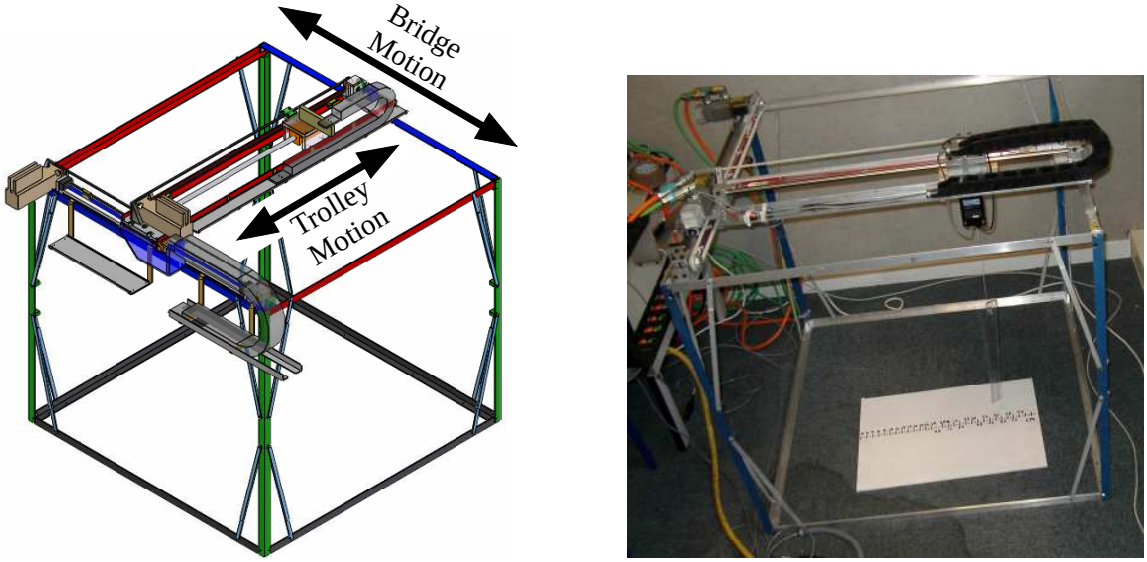


Figure 3.6: PLO Program Structure

The HiBay crane had three axes: bridge, trolley, and hoisting. To control the bridge and trolley axes simultaneously, two data blocks were required to store the velocity setpoints for each axis. Consequently the synchronous loop actually read values from both data blocks, and sends the corresponding velocity setpoint to the corresponding drive; bridge or trolley. Additionally, the bridge axis is actuated with two motors, one on each side of the bridge beam. To achieve coordinated motion between the two motors they were connected in parallel to the single, bridge drive. The hoisting axis was not controlled with the PLO, it was controlled directly from the pendant using the existing relay system.

3.3 Bridge Crane

The goal of the bridge crane project was to build a portable, programmable bridge crane capable of performing input shaping experiments and being used as an educational tool. It needed to be portable because it was shipped to GT Lorraine in France, and had an integral role in the controls courses taught there. A sketch of the bridge crane is shown in Figure 3.7(a) and a picture is shown in Figure 3.7(b). Its size is approximately 1 m^3. It is capable of movement in both trolley and bridge directions. Later work by John Hued and Josh Spiers added a hoisting axis.



(a) Bridge Crane Sketch.

(b) Bridge Crane Picture.

Figure 3.7: Bridge Crane Experimental Testbed: Sketch and Picture.

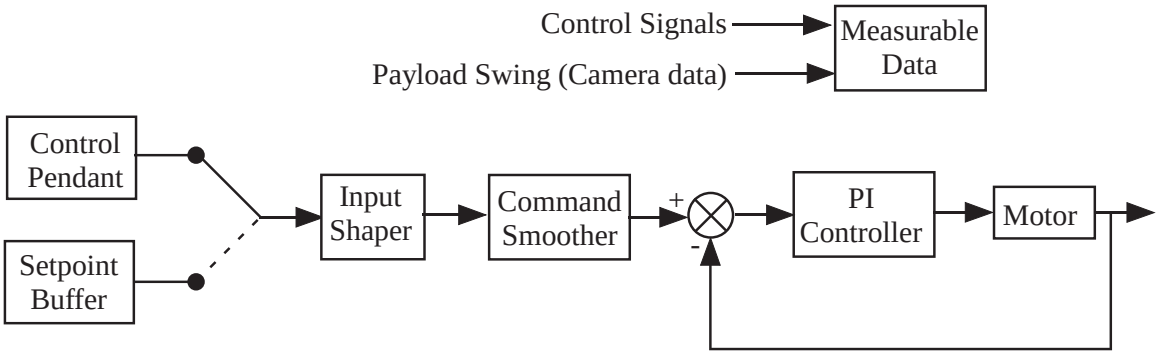


Figure 3.8: Bridge Crane Block Diagram.

Figure 3.8 shows the electrical layout of the crane. This layout is similar to the Hibay crane. The control pendant has four directional buttons to drive the crane and two buttons to change the controller mode. The pendant signal is sent to a Siemens PLO, the PLO applies the input-shaping algorithm, and generates a series of velocity setpoints for the drives. The user also has the option to run a trajectory stored in a setpoint buffer. The velocity setpoints are then interpreted by the drives which, in turn, send the power signal to the motors. Unlike the HiBay crane, the crane uses synchronous AC servomotors with encoders. The encoders provided accurate trolley position data during experiments and are also used to close the Proportional-plus-Integral velocity feedback loop in the drives.

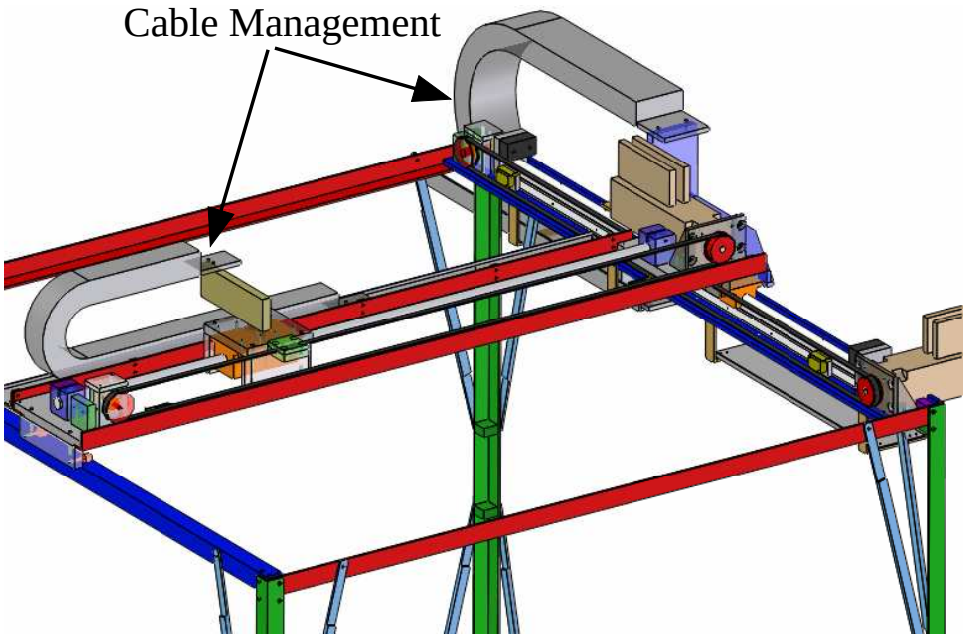


Figure 3.9: Bridge and Trolley Axes.

While the bridge crane and HiBay crane shared a similar electrical system, the crane also had several new features that proved challenging to design. The entire physical structure and actuation system need to be portable for shipping overseas to France. The bridge crane was also required to use a vision system to measure the payload swing.

3.3.1 Axes Design

The crane structure was designed in Solid Edge, a CAD modeling program. The final solid model is shown in Figure 3.7(a). The edges of the cube structure were all made from 1" aluminum angle stock. This material was chosen because it is cheap, light, and easy to machine. Small aluminum braces were added at each corner to provide extra structural stability and minimize structural vibration.

At the heart of the crane design were the bridge and trolley axes. A close-up of these axes is shown in Figure 3.9. Both axes were belt driven directly from the motors. Belts were chosen because they were cheap, light, and relatively easy to install. Timing belts were used to minimize slippage and backlash. Timing belts had the added advantage of only requiring minor tensioning. Another design feature was the cable management system, shown as the C-shaped grey tubes in Figure 3.9. These fed the power and encoder cables

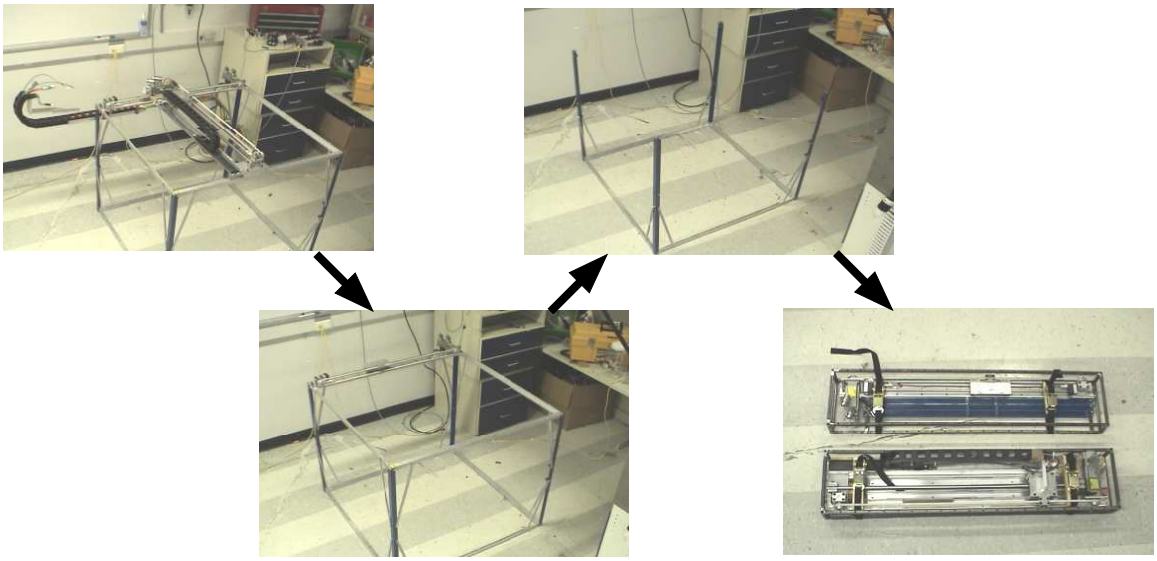


Figure 3.10: Bridge Disassembly Picture.

to the motors and camera while the system moved. The plastic tubes bend in a uniform, repeatable way so that the cables did not become tangled or stressed.

3.3.2 Design for Portability

A big design challenge for the crane was making the system portable. The crane was designed to be light and compact, to make it easy to transport. Almost all of the structural elements are lightweight aluminum and made as thin and small as possible without degrading the structural integrity. In addition, the structure was designed to be easily disassembled and collapsed to fit into a compact traveling box. Pictures of the actual crane being disassembled are shown in 3.10. All of the disassembly can be performed with a single socket-head wrench.

3.3.3 Vision System

An important feature of the crane was the vision system. Figure 3.11 shows two designs that were implemented; the initial design is shown on the left and the final design is shown on the right. The objective was to use a digital camera to measure the payload swing. The digital camera was produced by DVT Inc. and designed for object recognition and tracking in industrial environments. In the initial design, the camera was fixed to the trolley and

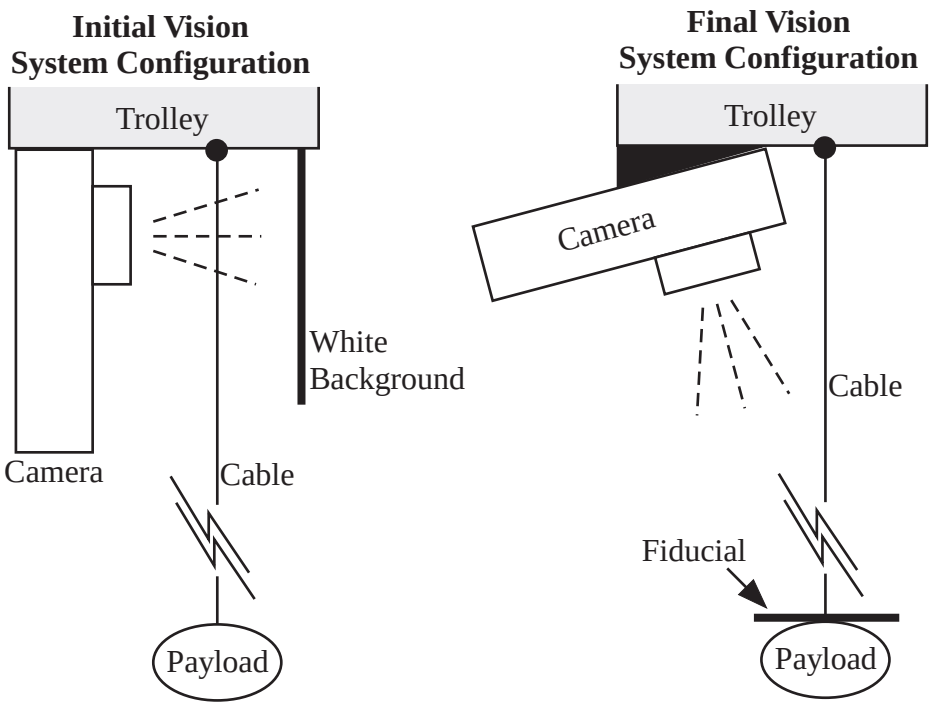


Figure 3.11: Vision System Designs.

pointed orthogonal to the payload cable. A white background was placed behind the black cable to improve the image contrast for the camera. The camera took pictures every 100 ms. In that time the camera would identify the edge of the cable, fit a line to it, and output its deflection angle from vertical. All this analysis was performed realtime in the camera using its internal microprocessor.

Although the initial design successfully measured the cable deflection, it had several drawbacks. To measure the cable deflection in two directions would require two cameras, which added weight and cost. Since the camera was so close to the cable, a costly lens was need to focus the image. The close proximity of the camera to the image also gave the camera a limited field of view, which limited the range of measurable angles. Finally, it was observed that the camera sometimes had trouble identifying the cable due to inconsistent image contrast.

To solve these problems a second vision system design was employed, shown in the right of Figure 3.11. In this design the camera points down at the payload and tracks a fiducial. The fiducial is a circular marker made of retro-reflective paper that was fixed to the top of

the payload. Because the retro-reflective paper is so much brighter than the background, it is easy for the camera to identify and track it. In this configuration the camera was programmed to search each image for the circular marker, and output the coordinates of its center point. The camera cycle time is 40 msec. The advantages of this new design were:

- Only one camera could be used to output two dimensional payload coordinates
- The image was far enough away that standard, inexpensive lenses could be used. It also gave the camera a large field of view to measure a wide range of deflection angles.
- The retro-reflective paper provided consistent contrast that made it easy to identify the marker.

Much of the design and implementation of the final vision system design was performed by Khalid Sorensen and John Hued.

3.4 Tower Crane

The goal of the tower crane project was to build a portable crane that could be used for research and education. The tower crane was shipped to the Tokyo Institute of Technology in Japan. The tower crane had several added features, which the previous cranes did not have:

- A graphical user interface (GUI) was developed to control the system from the PC.
- The whole system was configured to be controlled over the internet.
- A hoisting mechanism was designed.
- An anti-swing feedback control algorithm was implemented so the payload could be steadied by remote users.

These design challenges, and their solutions, are described in depth below.

A sketch of the tower crane is given in Figure 3.13 and a picture is given in Figure 3.12. The crane is approximately 2 m tall with a 1 m jib (arm). The crane has 3 degrees of freedom actuated with Siemens synchronous, AC servo motors. The slewing motor controls

Figure 3.12: Tower Crane Picture.

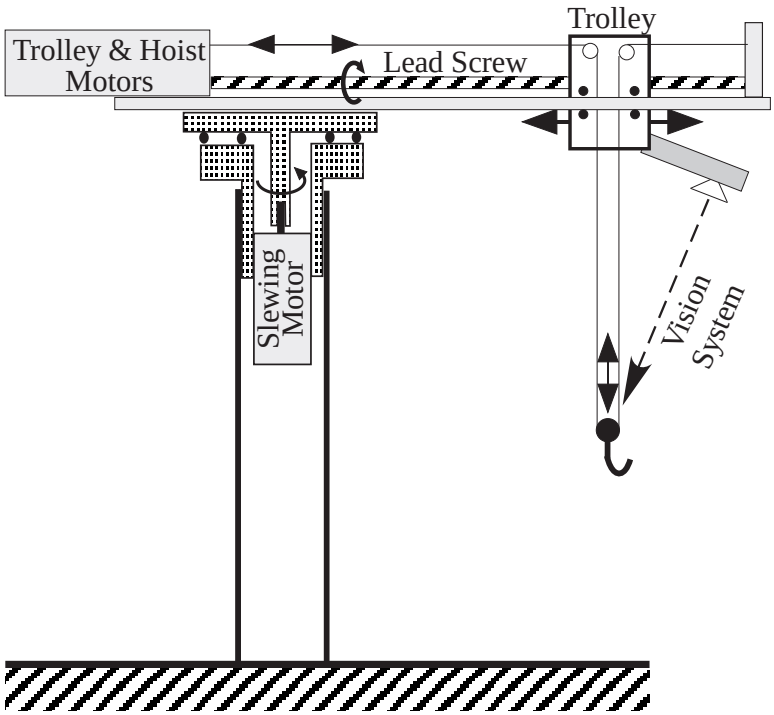


Figure 3.13: Tower Crane Sketch.

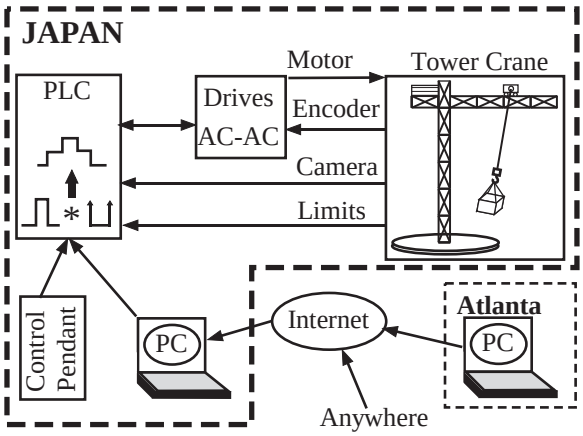


Figure 3.14: Tower Crane System Overview.

the rotation axis, which is capable of 340° rotation. The trolley motor moves the trolley radially via a lead screw. The hoisting motor controls the suspension cable length. In addition, a Siemens digital camera is mounted to the trolley and records the swing deflection of the payload, at a rate of 40 msec.

Figure 3.14 gives an overview of the electronic layout of the tower crane. A control pendant or PC can be used to drive the tower crane. Each sends the control signal to a Siemens PLO, which sends the velocity setpoints to the motor drives. In addition, the PLO receives and stores the payload swing data from the camera. The motors are powered with Siemens Sinamic drives, which use the motor encoder signals to provide Proportional-plus-Integral (PI) velocity control of the motors.

3.4.1 Graphical User Interface (GUI)

Figure 3.14 shows that the PLO can receive velocity commands from either a control pendant or a PC. The PC controls the crane using the on-screen graphical user interface shown in Figure 3.15. The upper left portion of the screen shows a real-time animation of the crane configuration from an overhead view. The square is the trolley position and the circle is the payload position. The current configuration is also numerically displayed in the bottom center of the display (slew angle, trolley pos, etc ...). The crane can be manually driven using the directional arrows at the bottom left of the screen. In addition, velocity setpoints can be pre-programmed and then executed with the "Play" button. Other features include;

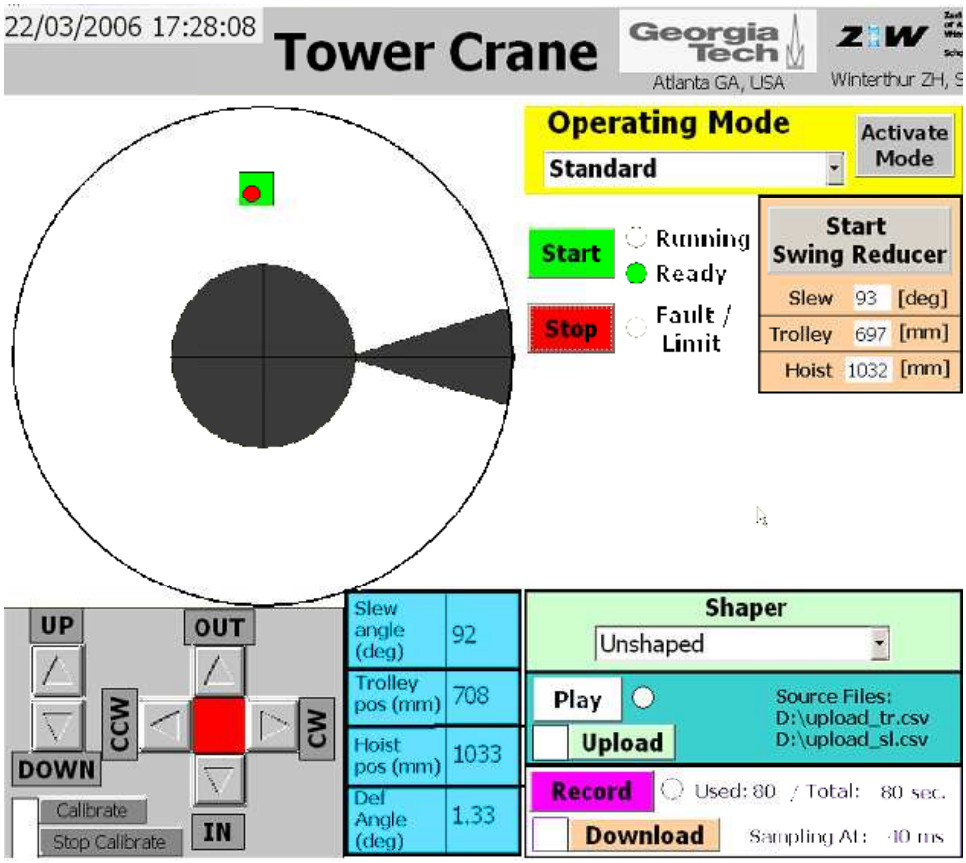


Figure 3.15: Tower Crane Computer Interface.

input shaper selection, data record, and a "swing reducer" which uses feedback from the camera to automatically damp out payload sway.

3.4.2 Internet Control

As mentioned earlier, internet control was an important aspect of this project. To achieve this goal, the controlling PC was equipped with UltraVNC. This program allows any user with internet access to remotely control a target PC. Students from anywhere around the world can access the GUI interface and run the crane, as shown in Figure 3.14. The tower crane resided in the Tokyo Institute of Technology in Japan during most of its operation. There it was remotely controlled by students and researchers at Georgia Tech, the University of Tokyo, the University of Buffalo, Korea, and Switzerland.

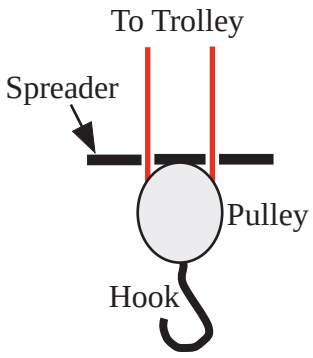


Figure 3.16: Cable Spreader on the Hoisting Pulley.

3.4.3 Hoisting

A detailed sketch of the hoisting mechanism was shown in Figure 3.13. Notice that due to the way the cable is threaded, the hoisting motor does not need to be located on the trolley. This lowered the weight of the trolley significantly and helped balance the weight of the jib. This hoisting configuration is similar to the ones found on real tower cranes.

One of the challenges with this design was payload twisting. If the payload struck an object it would sometimes impart a moment that would cause the payload to start spinning. As a result the cables would twist, preventing them from feeding through the pulley that held the payload. This would cause problems for hoisting and trolleying motions which force the cables through the pulley. The solution was to use a cable spreader, shown in Figure 3.16. The cable spreader kept the cables separated as they left the pulley. The cable spreader was a thin plastic square mounted to the top of the pulley. The cable was threaded through two holes in the plastic on either side of the pulley. If the pulley did start spinning, tension from the cable spreader and the pulley weight would quickly damp out the rotation. The flat surface of the cable spreader also provided convenient location to mount the payload fiducial for camera tracking.

3.4.4 Anti-Sway Controller

An anti-sway feedback controller was implemented on the tower crane. The purpose of this controller was to make it easier for remote users to steady the payload. A block diagram of the anti-sway controller is shown in Figure 3.17. Similar types of algorithms have been

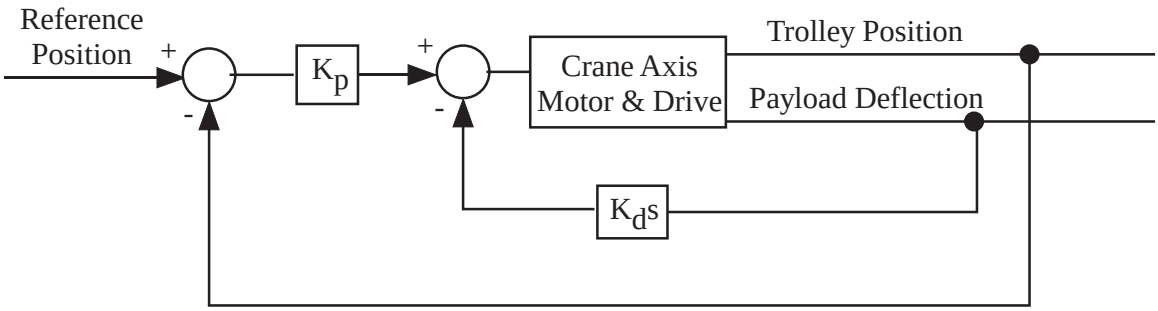


Figure 3.17: Anti-Sway Feedback Control Loop for Trolley and Slewing Axes.

developed by Sorensen [95–97] and Hillsey [27]. This control law was implemented on both the trolley and slewing axes. The controller consists of two feedback loops, as shown in Figure 3.17. The outer loop controls the position of the trolley using position feedback. The controller operating on the outer loop is a simple proportional control. The inner feedback loop damps out the payload swing using the measured angular deflection of the payload. A derivative controller was chosen for the inner control loop to counteract three sources of phase lag not shown in the figure:

- Rise time of the motors.
- Delay filter on the deflection measurement to filter out noise.
- Control loop digital cycle time.

The phase lead from the derivative control balanced this lag and yielded a stable controller.

The anti-sway controller is activated from the GUI shown in Figure 3.15. When the controller is activated it uses the current trolley position as the reference position in Figure 3.17. As a result, the controller immediately begins damping out any payload vibration while also trying to maintain the current trolley position. While the swing reducer is activated the user can also type in any desired 3-D coordinates (trolley position, slew position, and hoisted height), in the fields beneath the "Start Swing Reducer" button. These coordinates become the new reference position in the feedback loops and the crane moves to the new position with a very low level of swing. A simple position control loop was used to control the hoisted height (not shown in the figure).

3.5 Conclusions: Lessons Learned

All three crane projects were a success. They met the design objectives and became valuable assets for this thesis. They were used in universities around the world to teach students. The systems are robust enough that they continue to be used by researchers and students to this day. These setups have been used in 13 papers [8, 13, 29, 30, 33, 34, 47, 48, 53, 86, 95–97], to date.

In the course of constructing and building these devices several lessons were learned, which are discussed here. The focus of this thesis is not design, so this discussion does not propose any new discoveries in design theory or practice. The purpose of this discussion is to help any future designers that would construct similar devices so that the knowledge gained is not lost.

3.5.1 Modeling and Layout: 3-D vs. 2-D Computer Design

Constructing the tower and bridge cranes from the ground up offered a rare opportunity to try two different design methods. For the bridge crane, the entire design was modeled using a SolidEdge, a 3-D solid modeling package. However, the entire tower crane was designed using a simpler 2-D drawing program, called Canvas. The original reason for switching to the 2-D drawing program was to speed up the design process, and because it was thought the 3-D modeling was unnecessary for such a simple machine.

Comparing the two design methods revealed that each technique had its advantages and disadvantages for each stage of the design process. In the early stages of the design process the 2-D program was much easier to work with than the 3-D program. During these early stages the design was very fluid and would undergo large changes in a small amount of time. Making large changes in the 3-D program was difficult because of all the steps necessary to create and locate a part in 3-D space. If objects were referenced to each other even a small change to single object, could create drastic changes to the objects around it. For example, consider a motor, a limit switch, and a spacer that are all position-referenced to a square mounting plate. If the mounting plate needs to be exchanged for a new part, a circular mounting plate for example, all of the other parts (the motor, limit switch, and

spacer) need to be re-attached to the new part in new ways. For the 2-D program design modifications were easy; objects could simply be dragged around, inserted, or deleted. In the later stages of the design, the 3-D package became more valuable. The 3-D package was better at capturing all the details of the design and checking for part interference. It also had built in functions for creating machine drawings from the modeled parts.

The aforementioned observations seem to indicate that the ideal process is to use the 2-D drawing package for the early stages of the design, and the 3-D package for the final stages of the design. However it is still unclear when this transition should take place. It depends, in part, on how skilled the designer is with each program. A designer skilled in 3-D modeling might know how to create models that are more robust to design changes. In this case the transition to the 3-D modeling could happen earlier. On the other hand, a designer who can render good detail in the 2-D program in little time might transition to the 3-D program later. Some discussion of this can be found in [68].

3.5.2 Portability

Both the bridge and tower cranes were designed to be portable. The original specification was to make it possible for these devices to be assembled in under 30 min. In both cases this design spec. was found to be overly conservative since there was usually at least a day to get the system up and running. In addition, while some parts were easy to disassemble and reassemble, others were very difficult to service. A more realistic constraint would have been design for assembly in 6 hours with an emphasis on *easy assembly* not portability. Easy assembly refers to all parts in the assembly. An easily assembled device would not only be portable, but also easy to service and modify if necessary.

3.5.3 Material Selection

The bridge and tower cranes needed to be light, yet strong. The structure needed to be light to reduce shipping costs and achieve rapid dynamic response. However, it also needed to be strong enough to support its own weight and not vibrate during motion. Most of the parts in both designs were made from aluminum due to its excellent strength-to-weight properties and easy machinability. However, many of the parts still ended up being bulky

and heavy. Plastic was used in the tower crane in an attempt to alleviate this problem. However it was found that in most scenarios thick plastic was needed to give the desired strength, which made parts unacceptably heavy.

Future designs might consider sheet metal. It is light, inexpensive, and easy to machine, stamp, or bend. It can also be made strong by bending and/or assembling pieces into different shapes. For example, future designs could use sheet metal with a truss-like structure stamped or machined out of it. These pieces could then be assembled into triangular or rectangular prismatic beams, similar to those found on real tower cranes. Another alternative structural material is composite tubing. It is extremely light and strong and has been used on several robotic machines.

3.5.4 Choice of Actuators

The choice of actuators were found to have a tremendous impact on the design. For example, in both cranes the motors were the single heaviest part in the whole system. This had a direct impact on the size and weight of the system, which influenced the shipping costs. It also increased the size and weight of the structural design. Another impact was from the cables. The motors used in the crane projects used 1/4”-1/2” diameter cables. These added more weight and necessitated the use of a cable management system, which also increased the size and weight of the cranes.

CHAPTER IV

CURRICULUM DEVELOPMENT

Input shaping is a growing field in engineering. Its uses can be found in thousands of machines all over the world, from cranes to disk drives to satellites. Not only is it extremely effective, but it is one of the easiest control strategies that exist. From an educational standpoint it is also a very practical and intuitive demonstration of several basic control concepts such as vibration, convolution, command design, etc... Unfortunately, few universities teach it in their controls curriculum. This section discusses new educational tools and methods that were developed to teach input shaping in a dynamics and controls curriculum. This section also discusses how these new tools were implemented in universities around the world with great success.

4.1 Advances in Dynamics and Controls Curriculum

The advances in dynamics and controls curriculum were designed to meet several goals. Specifically we wanted the students to:

- Learn input shaping theory
- Work with industrial grade components and learn how to program, take measurements, and perform experiments with these components
- Get a hands-on feel for oscillatory systems and input shaping through operation of a real crane system.
- Design and program input shapers into a real system. Then, perform experiments to validate their design.
- Work on international collaborative projects.

These goals are aligned with the Accreditation Board for Engineering and Technology (ABET) guidelines discussed later. Notice that there is particular emphasis on working

with real-world machinery. This not only gives the students experience with working on industrial equipment, but also makes the theoretical concepts more meaningful. To achieve this goal several laboratories were developed that used the HiBay crane, bridge crane, and tower crane. These cranes were discussed in chapter 3. The focus on incorporating labs and real-world experience also corroborates with the ABET guidelines.

4.1.1 Core Concepts

The focus of this study was teaching input shaping. The reason for teaching input shaping is because it is an excellent tool for teaching basic engineering concepts. These core concepts include:

- System Vibration
- Convolution
- Superposition
- Robustness

By learning and working with input shaping concepts, the students obtain insight and knowledge about these core concepts. Input shaping shows these concepts at work and applies them in a very practical way. All of the lab assignments and lectures emphasize one of these four concepts.

4.1.2 Educational Milestones

In pursuit of these goals, several milestones were reached. These milestones are shown in the time-line in Figure 4.1 as phases I to IV. The figure describes the major accomplishments of each milestone, as well as what classes used these new educational tools. The timeline begins with some background. In 1999 Dr. Singhose first introduced input shaping into the Georgia Tech curriculum. Following that Forest, et. al. [21] developed a computer simulation of a input shaped crane so the simulated crane could be driven real time from the computer. It was then incorporated into the input shaping curriculum through homework assignments.

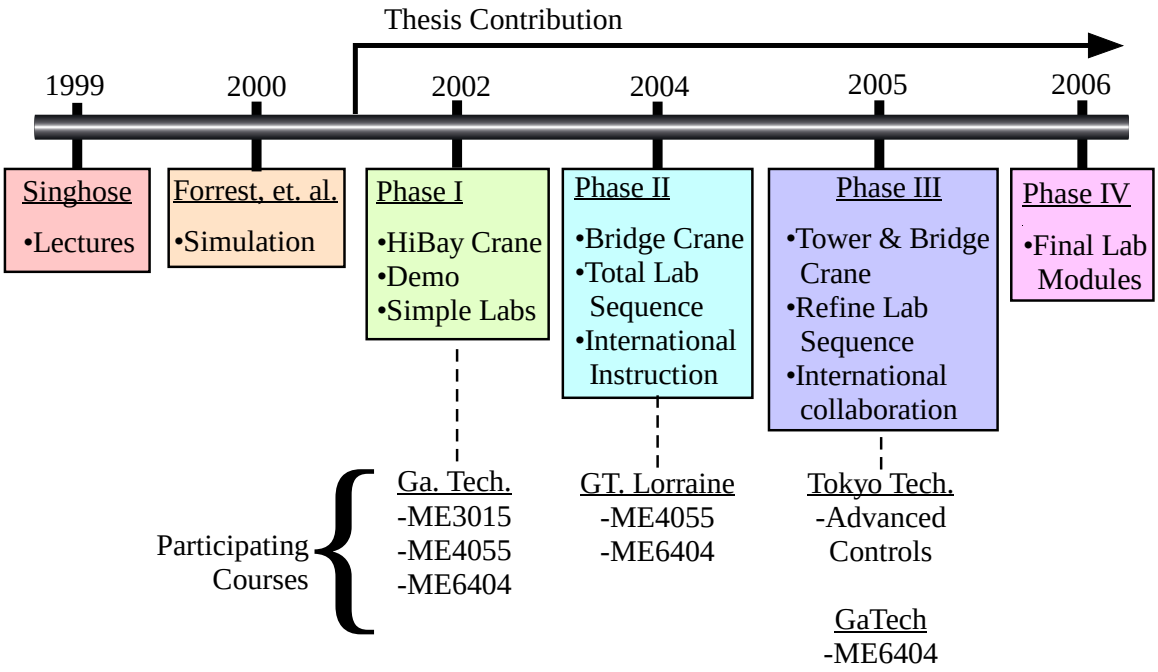


Figure 4.1: Time-line of Educational Milestones

The students would drive the crane with and without input shaping to get a feel for how the algorithm worked.

My contributions to input shaping education began in 2002, labeled phase I in the Figure. The preliminary phase developed the input-shaped HiBay crane. It was then used in three different classes at Georgia Tech at both the graduate and undergraduate level. Students had the opportunity to drive the crane, measure its response, and design and implement their own shapers. The crane has since been used in numerous classes and research projects. It has been the subject of 5 research papers [33, 34, 86, 95, 96].

In 2004, phase II, the bridge crane was completed and shipped to Georgia Tech Lorraine (GTL) in France. A complete lab sequence was designed using the crane to teach input shaping. This sequence was the first stage in developing a finalized set of lab modules for input shaping. This phase culminated with a Siemens conference at GTL in which the students presented research projects involving the bridge crane. The portable bridge crane continues to be used in classes, as described below, as well as input shaping research [47].

In 2005, phase III, the tower crane was completed and shipped to Tokyo Tech in Japan. Both the tower crane and bridge crane were used as part of a joint graduate controls course

simultaneously taught at Tokyo Tech and Georgia Tech. Students not only completed the input shaping lab sequence, but also collaborated as international teams to complete term-projects that were published in three conference papers [8, 13, 30]. Using the internet students could remotely control both cranes and worked together in international teams. In 2006, phase IV, the input shaping lab sequence was refined into its final form. Since then, the portable tower crane has continued to be used by universities around the world for research and education including: Tokyo institute of Technology and Korea Institute of Technology. In addition, the tower crane is part of an ongoing, international, remote crane operator study involving hundreds of crane operators that originated from one of the student's projects.

Throughout the educational advancements, cranes are the experimental testbed used by the students. Cranes were chosen because their vibration is usually on a large scale (eg. inches or feet) that is easy for the students to see and measure. The vibration of other industrial equipment, such as pick-and-place robots or coordinate measuring machines, is typically on a much smaller scale (eg. microns). Measuring the vibration of these machines usually requires expensive equipment, such as a laser interferometer, and gives the students less of a hands-on feel for the system. Furthermore, crane systems are versatile*- and can be easily expanded to teach more advanced concepts such as multimode and nonlinear systems.

In summary, laboratory-based curriculum was developed and tested in seven courses, three different universities, across three different continents. The "large scale" of this effort has led to an extensive set of educational data and a well-tested curriculum. The rest of this section discusses these educational achievements in more detail. The next subsection reviews more background on the ABET standards for engineering curriculums, as well as what other universities have accomplished in terms of input-shaping curriculum. Then, the four phases cited in the timeline are discussed in detail.

4.2 Education Background

One goal of this thesis was to develop a set of labs that could be remotely performed from anywhere in the world. Remote laboratories are an active area of research. The number of

schools offering on-line or distance degree programs has been steadily growing [12]. Initially, these were designed for students with part or full time jobs getting their master's degree. As these online programs expanded, their aim shifted to include undergraduate programs as well. However, undergraduate engineering programs usually include a laboratory that requires students to physically be in the lab. This started raising questions [20, 24] : Can labs be performed remotely? Does the student get a different experience doing the lab online or in person? If a lab could be done online, why not use a simulation rather than invest in real equipment?

The issues surrounding online labs became paramount when universities started seeking ABET accreditation for their online undergraduate programs. ABET did not have any standards in place for on-line labs [20, 72]. In fact, ABET realized that there were no standards for laboratories in general. To answer these questions a colloquium was created and funded by the Sloan Corporation in 2002 to answer these questions. The colloquium failed to establish rigorous standards for labs, online or otherwise. However, 13 broad objectives were created to serve as a framework for creating labs [12, 20]. In addition many of these objectives could easily be achieved by a remote lab.

Despite the challenges, more universities are creating on-line labs. A list of on-line labs for 2002 is given in [12]. The Sloan Corporation, that sponsors many online programs, recognizes that remote labs add to the "quality and scale" of online courses [12]. New research is also suggesting that online labs have as much value as standard labs. In a study by Leydens et. al. [57] a group of students performed a fluids lab online, while the other group performed the same lab in-person. The study found that there was no significant difference between the grades of the two groups.

Input shaping is also becoming incorporated into engineering curriculums throughout the world. Figure 4.2 shows an overview of universities teaching input shaping, and the state of their curriculum. Notice that there are several major engineering universities both in the US and around the world that are teaching input shaping. Most only teach the theory in lectures and then have the students run simulations. One university, the Tokyo Metropolitan Institute of Technology, has created a demonstration device and developed

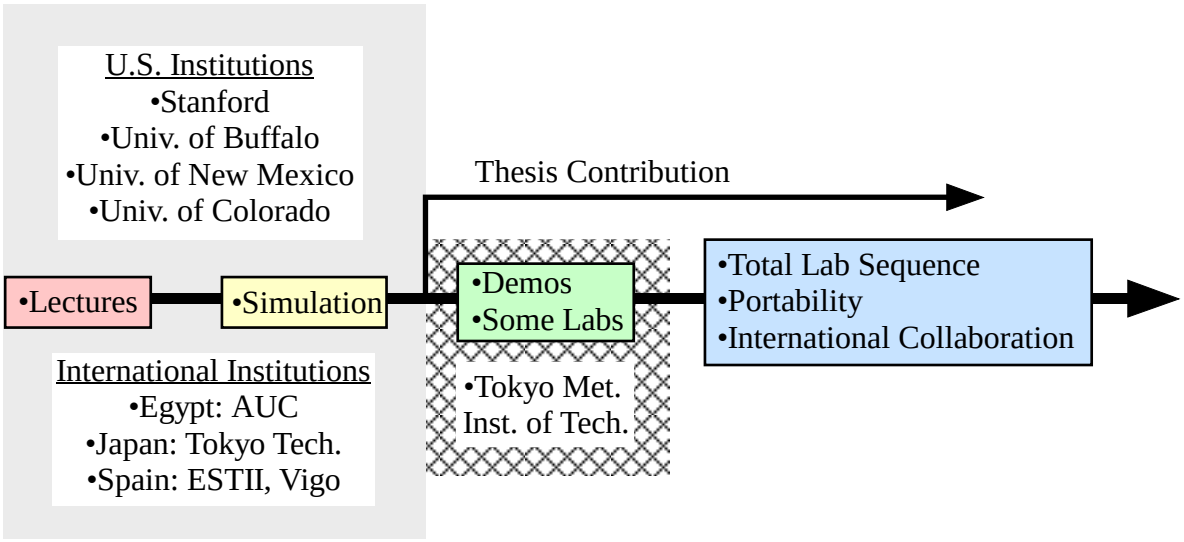


Figure 4.2: Overview of Universities Teaching Input Shaping.

laboratory activities. However, Georgia Tech has a very advanced input-shaping curriculum with a full lab sequence using several experimental devices on an international scale.

The emphasis on hands-on labs is motivated by the way that students learn. Bloom's taxonomy is frequently cited as a model of the learning process [9]. However, applying this model to engineering education is challenging and a better model of the learning process would be [83]:

1. Exposure
2. Familiarity
3. Understanding
4. Ability to Apply
5. Ability to Innovate

Hands-on experience with real equipment helps the students to progress through these stages of learning. Demonstrations push the students through *Familiarity* and towards *Understanding*. Laboratory assignments require students to implement the concepts they learned in lecture which gives them the *Ability to Apply* what they have learned. Once the students understand the theoretical concepts and have the skills to operate the equipment,

they are given the freedom do research on their own in final projects. In these final projects students may even demonstrate the *Ability to Innovate*. All of these tools are implemented throughout the four educational phases of this thesis.

4.3 Phase I: Initial Curriculum Development at Georgia Tech

In the first phase of the education project, the input-shaped HiBay crane was used in graduate and undergraduate controls courses at Georgia Tech. The crane was used to teach basic input shaping techniques and give the students hands on experience with real control systems.

In System Dynamics and Control (ME3015), an undergraduate course, the crane was used as a demonstration device to teach input shaping. After the students learned the basics of input shaping in the classroom, they were taken to the Hibay and allowed to drive the crane with and without input shaping. This showed the students, first-hand, how input shaping improves crane response. They then used simulations in homework to design input shapers, measure its response, and drive a virtual crane through an obstacle course.

In Experimental Engineering (ME4055), the senior laboratory course, students designed various experiments using the crane setup. The students captured the crane-swing with a Camcorder and then used image processing to measure the swing with and without Input Shaping. This assignment was designed to give the students experience with advanced data-acquisition techniques for real world machines.

In Advanced Control System Design and Implementation (ME6404), the graduate-level controls course, students designed input shaping algorithms and tested them on the crane. The students learn the theoretical basis behind input shaping and then use this knowledge to design input shapers of their own. The student shapers are then programmed into the PLO and run on the crane. The shapers are tested for their ability to suppress vibration and robustness to modeling errors.

4.3.1 Phase I Conclusions

Using the crane to teach input shaping achieved many goals. The students got hands on experience using a massive dynamic system and performing experiments. They also got the opportunity to design input shapers and implement them on a real system, rather than simply in a simulation. Another successful part of these classes was group-work. By working in groups the students learned communication and teamwork skills. The students also helped each other solve problems and learned from one another.

The experience also generated many observations that paved the way for the next phase. For instance, due to safety concerns and the size of the crane, the students had limited freedom to adjust the control system. A smaller machine would eliminate these concerns. It also became apparent that for the students to fully explore the system and perform experiments they would need to be more fully trained on how to operate the system. Finally, in all three classes the crane only made up a relatively small portion of the class since this was the first time this system was being introduced. In the next phase, the laboratory activities would be more extensive.

4.4 Phase II: Curriculum Development at GT Lorraine

The portable bridge crane was used in two controls courses at Georgia Tech Lorraine (GTL). The graduate students performed six labs using the portable crane that were intended to reinforce many of the concepts they learned in lecture. Figure 4.3 shows a block diagram that summarizes how the portable bridge crane was configured for these labs. Notice that the students had access to four main blocks in the control architecture: the ramp generator, the PI controller, the input shaper, and the trajectory buffer. In addition, the students measured the payload swing either manually, with a ruler, or electronically using the digital camera. They could also measure any of the control and/or feedback signals in the control loop.

Figure 4.3 also shows how the lab sequence was structured. Each lab introduced the students to a new block in the system. The students adjusted parameters in each block and measured its effect on the response. By the end of the lab sequence the students understood

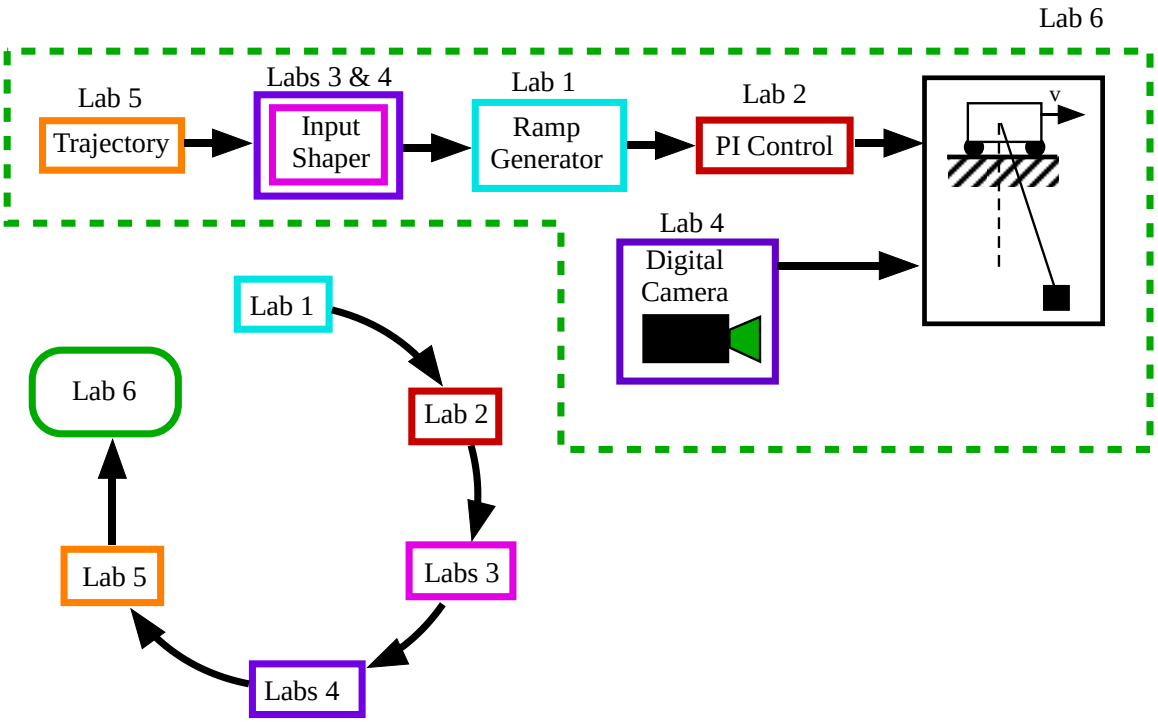


Figure 4.3: Lab Assignments Overview

how to control every aspect of the crane. The lab assignments also taught the four core concepts discussed earlier: system vibration, convolution, superposition, and robustness. As a final assignment, the students were asked to demonstrate their knowledge of the system and the concepts taught in class by developing their own final projects. The result of was wide range of creative studies that included experimental results from the crane. Their work was presented at a Siemens conference at GTL.

4.4.1 Lab Descriptions

Lab 1 *Goals:*

- Get students acquainted with the equipment.
- Observe how the rise and fall times of the ramp generator, shown in Figure 4.3, effect the motor response.
- Observe how the rise and fall times of the ramp generator effect payload swing.

Procedure: A simple point-to-point move is preprogrammed into the PLO. The students use a drive parametrization program to adjust the rise and fall times of the ramp generator. These parameters effect how quickly the drive responds to the PLO commands during the point-to-point motion. The students measure the motor response with encoders and by visually observing the response. The students also manually measure the effect of changing the rise time on the payload response using a ruler.

Results: Increasing the rise time causes a sluggish motor response. However, this makes the payload oscillate less. The students learned basic operation of the crane including how to program the ramp generator. This lab also acquainted them with working in groups and writing up their results in a lab report.

Core Concepts: The core concept emphasized in this lab is system vibration. More specifically the lab targets the relationship between rise time and oscillatory response.

Lab 2 *Goals:*

- Tune the PI controller gains to achieve small trajectory tracking error.
- Tune the PI controller gains to achieve small payload oscillation.

Procedure: The students manually adjust the proportional and integral gains in the drive. The students measure the effects on the motor response and the payload response.

Results: Increasing the PI gains will improve tracking error. However if the gains are set too high the system goes unstable. Choosing lower PI gains reduces payload oscillation because it causes a sluggish system response. The students learned how to program the PI controller and also how to tune a PI controller based on system response using the aforementioned concepts.

Core Concepts: The core concept emphasized in this lab is system vibration. The students learned that a controller can be tuned to have a rapid response, but this may induce vibration.

Lab 3 *Goals:*

- Program a simple input shaper into the PLO.

- Compare shaped and unshaped payload swing for various move distances.
- Compare shaped and unshaped payload swing for various payload suspensions lengths.

Tasks: The students design a simple input shaper for the baseline pendulum length and enter it into the PLO. The students compare the payload swing for a point-to-point motion with and without input shaping. The experiment is repeated for various move distances and cable lengths. The results form a sensitivity analysis of the shaper.

Results: Input shaping reduces the payload swing in all cases. However, for some move distances the unshaped motion also creates "natural input shaping" that causes low levels of swing. In addition, changing the payload length degrades input shaping performance because it changes the system natural frequency. The students learned how to design a simple ZV shaper and program it into the system. They also learned about shaper sensitivity and measured it for a ZV shaper.

Core Concepts: The core concept taught in this lab is superposition. In class students learn that input shaping eliminates vibration through the superposition of vibratory response. In this lab the students get to see this concept first hand and measure the results.

Lab 4 *Goals:*

- Create and test a robust shaper, a fast shaper, and a multimode shaper.
- Understand the strengths and weaknesses of each shaper.

Tasks: The students create a variety shapers using the known system parameters and the concepts they learned in lecture. The system response is measured using a digital camera that automatically calculates the swing angle every 100 ms. For the robust shaper, the students measure the payload swing for various pendulum lengths and compare the results to the previous lab. For the fast shaper, the students measure the response time and also measure the swing for one and two-mode payloads. For the multimode shaper, the students perform an FFT analysis on the two-mode payload that yields the system natural frequencies. Then, they design a multimode shaper using a combination of the single-mode shapers and test it on the two-mode system.

Results: The students learn about and measure the strengths and weakness of a variety of shapers. In the process they realize that robust shapers are slow and fast shapers excite high frequency modes. They also learn how multimode shapers can be constructed from simple one-mode shapers.

Core Concepts: The core concepts taught in this lab are convolution, superposition, and robustness. Similar to the previous lab, input shaping works using superposition and the lab reinforces these concepts. To program the shapers for this lab and design successful shapers the students are required to perform convolution by hand. The lab is also designed to show the students the performance of shapers with different robustness. The students learn the advantages of robustness as well as the tradeoffs.

Lab 5 *Goals:*

- Simulate how to use input shaping in an industrial environment (eg. warehouse, shipyard, etc...).

Tasks: Generate an input-shaped trajectory to move a payload through a cluttered workspace. The same trajectory must be used for two different multi-mode payloads. The trajectories are generated in Matlab, downloaded to the PLO, and then run on the portable bridge crane using a real obstacle field. The trajectories are evaluated on task completion time and number of obstacle collisions.

Results: The students use many of the concepts learned in previous labs to find the right compromise between robustness and speed. The students also learn how to generate obstacle avoidance trajectories and optimize these trajectories for speed.

Core Concepts: The core concepts taught in this lab are convolution, superposition, and robustness. These three concepts are taught in the same manner as the previous lab. However, this lab presents the students with an open-ended problem and forces them to apply the concepts with less guidance than the previous lab.

Lab 6

The last lab requires the students to design their own, unique dynamics and control experiments. This forces the students to use combine their previous experiences with what

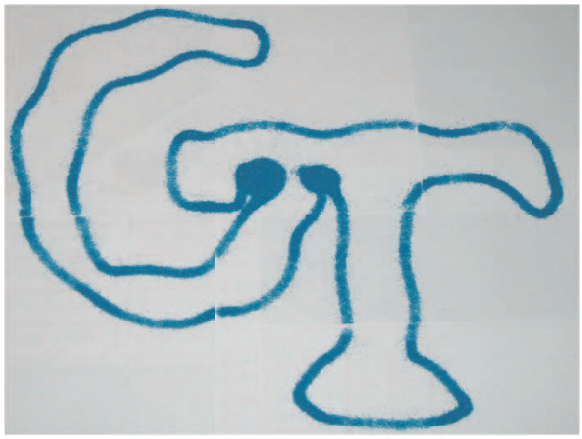


Figure 4.4: Sample Student Project: Pouring Out Sand to Spell "GT".

was taught in class. They must not only understand the ideas, but synthesize new ones. For example, one group used input shaping to eliminate multimode vibration for hoisting beams that can rotate. Another group used the crane to write "GT" in sand, as shown in Figure 4.4. These projects were high quality research and were presented in a controls and automation conference at GT Lorraine.

Core Concepts: This lab emphasizes all four core concepts: vibration, convolution, superposition, and robustness. The students apply what they have learned about these concepts from the previous labs to new and unrelated problems.

4.4.2 Undergraduate Laboratory Course

A second, undergraduate laboratory course was also taught at Georgia Tech Lorraine. The goal of the course is to teach basic laboratory procedures and writing methods. In this course the students were given the portable bridge crane as their lab device, as well as several options for projects they could work on. The students chose a project that involved testing a new input shaper designed to compensate for nonlinearities. Note that this is not a "mock" research project, the students dealt with a project that was (and still is) on the cutting edge of input shaping research. In their project the students:

- Programmed the digital camera to measure the payload position.
- Estimated the error of their sensors.

- Ran several experimental trials.
- Analyzed the data to determine the effectiveness of the new shaper.
- Wrote a formal lab report and gave a presentation detailing their work at the Siemens GTL automation conference.

4.4.3 Phase II Conclusions

The second phase of curriculum development successfully implemented many improvements over the first phase. A versatile, small-scale crane was built to expand the variety of educational opportunities and give students more freedom to adjust the controller. A full lab sequence was developed that taught a range of concepts from system response to input shaping to trajectory planning. The students also completed final projects that let them explore cutting-edge research topics which were presented at the Siemens GTL automation conference.

The second phase also revealed some opportunities for improvement. The programming interface for the portable bridge crane was awkward. Separate programs were needed to control the drive, PLO, and digital camera. The lack of a hoisting mechanism required the students to spend a lot of time to manually change the payload lengths. Finally, the vision system was awkward to use and would stop working at random times. All of these issues would be addressed in the next phase.

4.5 Phase III: Curriculum Development at Tokyo Tech

In phase III, two parallel courses were taught at Georgia Tech and Tokyo Tech. The Georgia Tech students primarily used the bridge crane in Atlanta along with the phase II lab sequence developed for GT Lorraine. The Tokyo Tech students primarily used the tower crane in Japan with a new lab sequence described below. The students used the crane in three ways: to perform labs, to conduct and participate in a remote manipulation study, and to collect data for their final projects. The Japan course taught command generation and feedforward control, with an emphasis on real world systems. The course and labs focused on the same four core concepts mentioned earlier: system vibration, convolution,

superposition, and robustness. Because the two courses were taught in parallel, the U.S. and Japanese students had the opportunity to collaborate on the final projects. The results of this collaboration were so successful that they were able to be published as conference papers [8, 13, 30].

4.5.1 Laboratory Assignments

Five labs were conducted in the Tokyo Tech class. In a typical lab the students programmed the machine, ran a test, and then analyzed the results. The students performed the labs in groups of 2-4, however the first lab was performed alone to give each student experience running the machine. The students had 2 weeks to complete the labs and turn in a report of their work.

The labs were created with several objectives in mind so that the students would have a unique and interactive learning experience. These objectives are listed bellow. A recent ABET colloquium on laboratories established 13 objectives that should be universal to all labs [20]. The listing bellow also shows how the original lab objectives fit within ABET's framework:

- Students learn to use state-of-the-art equipment to control and measure tower crane response. This includes using drives, a PLO, and the appropriate computer software. (*ABET Objectives 1 & 13: Instrumentation and Sensory Awareness*)

- Students learn to design and implement input shapers on the tower crane. Experiments are performed to observe its effects first hand. (*ABET Objectives 5 & 13: Design and Sensory Awareness*)

- Students understand the strengths and weaknesses of input shaping and also understand the limitations of the underlying theory. (*ABET objective 2: Models*)

- Students work on teams in which they must communicate with each other to accomplish the final goal. In addition, each team must create a lab report documenting their work. (*ABET Objectives 10 & 11: Communication and Teamwork*

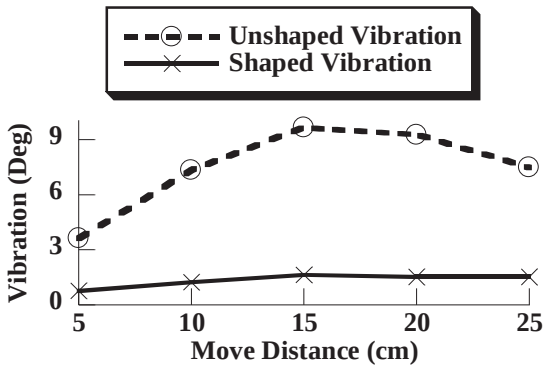


Figure 4.5: Student Data from Labs 1 through 4.

4.5.2 Labs 1 and 2: Characterizing the Unshaped Response

In labs 1 and 2 the students explored the basic oscillatory properties of the tower crane. In these experiments, the students moved the crane in a point-to-point motion and measured the vibration. The distance, velocity, acceleration, and suspension length were changed for the different trials to show how these variables effect the residual swing.

One of the fundamental concepts taught in this controls class is that the choice of command can have a profound effect on system response. This is related to the core concept of system vibration. Figure 4.5 shows one set of student generated data. The unshaped line gives the residual vibration amplitude for various move distances. Changing the move distance has a clear effect on the residual vibration, thereby demonstrating the above concept.

4.5.3 Labs 3 and 4: Input Shaper Design and Implementation

In labs 3-4 the students used input shaping to move the tower crane in point-to-point motion. Many of the trials from labs 1 and 2 were repeated, except with input shaping applied to the command. The objective was to show that an input-shaped command will reduce the residual vibration. In addition, the students learned about the strengths and weaknesses of different kinds of input shapers (eg. ZV [94], UMZV [88], ZVD [81]). Figure 4.5 shows the residual swing for various move distances using a ZV-shaped command. The plot clearly illustrates that input shaping reduces vibration and minimizes the dependance

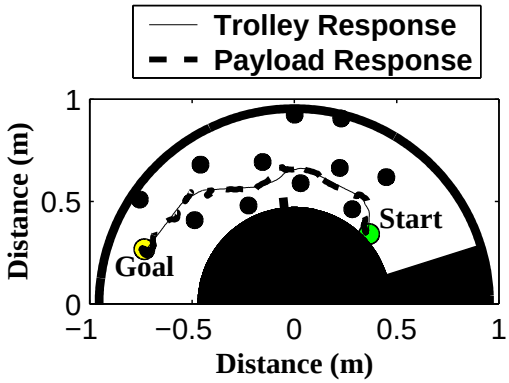


Figure 4.6: Student Data from Labs 5.

on move distance compared to the unshaped case. Similar to the GT Lorraine sequence, the core concepts taught in these labs are convolution, superposition, and robustness.

4.5.4 Lab 5: Crane Path Planning and Automation

In lab 5 the students were given the task of automating the crane to move the payload through an obstacle course. The groups competed to develop the best commands for moving in the shortest amount of time and with the fewest number of collisions. Figure 4.6 shows an overview of the obstacle course along with the start and goal points.

The design element in lab 5 forced the students to take the knowledge from previous labs and synthesize a solution to an open-ended problem. Figure 4.6 shows the results from one of the groups. Combining input shaping and intelligent path selection, the students were able to rapidly maneuver the payload through a tight path without collisions.

4.5.5 Student Final Projects

The U.S. and Japanese students worked in teams for the final projects. The students were given the option of choosing one of three projects:

1. Consider a crane where the trolley accelerates up to speed faster (or slower) than it can brake. Test a new input shaper developed for this system and develop an improved version of the shaper if possible.

2. Both tower and bridge cranes are governed by nonlinear dynamics. Given the equations of motion, verify these nonlinear dynamics and develop input shapers that try

and compensate for these nonlinearities.

3. Explore remote operation of tower and bridge cranes. Perform studies that test the effectiveness of input shaping when used to remotely control these cranes.

The full project descriptions appear in Appendix A: Japan/US Final Projects. Because the final projects were performed at the end of the class, the students were well trained on how to use both crane systems and collect data. In addition, since both cranes could be controlled remotely over the internet, the students in Japan or the U.S. could use either crane.

Due to the timing of the U.S. and Japan classes, the U.S. teams completed their final projects first. Then, the Japanese students were given the same projects along with the results from the U.S. students. Their task was to further develop the projects and improve the results from the U.S. students. In the final stage, the Japanese groups, and their U.S. counterparts, came together to write conference papers on their work [8,13,30]. The results for the third project appear in this thesis in Chapter 5.

In summary, the final projects were a great success. Rarely, if ever, do students get to work on cutting edge research and then publish their work as part of a engineering class. This opportunity provided significant motivation and ensured that the students developed a high-level understanding of the course material.

4.5.6 Follow-Up Test

To test the effectiveness of the teaching tools described above, a follow-up test was given to the students. The test was distributed to both Japanese and American students six months after the completion of the two courses. The test consisted of three questions shown in Appendix E. Each question focuses on one of the core concepts from the course: question #1 tests convolution, question #2 tests superposition, and question #3 tests robustness. Distributing the test several months after the completion of the course also tested how well the students retained the knowledge they learned. Of the 14 students that took the two courses, 11 completed the test. *Every student that completed the test got all three questions correct.* This is a clear indication that the teaching tools were not only effective, but that

the knowledge gained from this course was also retained by the students.

4.5.7 Phase III Conclusions

The third phase of the education project achieved many improvements over the second phase. A new GUI interface and program architecture made it easier for the students to use the system and program the controller. The hoisting axis made it easier to run different payload lengths. The new vision system was more robust and allowed a greater range of experimental tests. The students also completed final projects that were published as conference papers (the phase II final projects weren't published, only given as oral presentations).

4.6 Phase IV: Final Lab Modules

The culmination of this educational research was the formulation of a set of lab modules for teaching input shaping to a wide variety of students. These lab modules evolved over the course of the three phases already discussed.

The final lab modules were designed to maximize the strengths of the previous labs, while also addressing their weaknesses. The students were trained on how to use the equipment early in the lab sequence, so later labs could focus exclusively on the dynamics and controls concepts. The GUI was refined so it was easier to use. A large emphasis was also placed on giving the students a "hands-on" feel for using input-shaped systems. Each of these labs was aligned with ABET objects, in the same manner as was done for the Phase III labs. In addition, each lab focused on one or several of the core concepts discussed earlier: system vibration, convolution, superposition, and robustness.

Another important factor in designing the final version of these labs was remote operability. All of the lab modules can be performed remotely. In earlier phases this kind of functionality was not available. So the earlier labs had to be adapted so they could be performed remotely. For instance, in earlier versions of the labs the students would manually change the payload, say for double pendulum experiments. This was unpractical for remote labs so it was removed from the sequence. In addition, the GUI and lab procedures had to be improved to simplify the process for students controlling the lab remotely.

The end result of this process was a polished set of labs. The lab sequence is summarized below. The list also shows which ABET objects each lab targets [20]:

Module 0: Basic Operation Familiarize the students with the system. Students learn how to log on and drive the crane. They also become familiar with the graphical user interface (GUI). *ABET objective 1: Instrumentation*

Module 1: Velocity Variation Students move the crane without input shaping and measure the vibration. They observe that without shaping the payload oscillates and the amount of vibration depends on move velocity and move distance. This relates to the core concept of system vibration. This is also the first module where the digital camera is used to record response data. *ABET objectives 2, 3, & 4: Models, Experiment, and Data Analysis*

Module 2: Remote Obstacle Course Driving Students manually drive the crane through a simple obstacle course with and without input shaping. They observe first-hand that with input shaping the payload is much easier to control. Input shaping is a demonstration of the core concept of superposition, and the students observe this concept while driving the crane. This lab can also be used as part of a basic operator study to show how various crane operators (students) respond to the system with and without shaping. *ABET Objective 8 & 13: Psychomotor and Sensory Awareness*

Module 3: Input Shaper Design and Robustness Students design their own input shapers and implement them on the crane. They measure various properties of their shapers such as vibration reduction and robustness, one of the core concepts. In addition, designing input shaper teaches the core concept of convolution. *ABET Objectives 4 & 5: Data Analysis and Design*

Module 4: Automated Obstacle Course Navigation Students use trajectory planning and input shaping to automate the crane. They design a trajectory that moves the crane through the obstacle field in the shortest amount of time while avoiding

obstacle collisions. This lab emphasizes all four core concepts and requires the students to apply their knowledge to an open-ended problem.*ABET Objective 5, 6, & 7: Design, Learn from Failure, Creativity*

The full lab modules appear in Appendix B:Final Lab Modules. The tower crane is currently setup to perform all these labs remotely, and can be controlled from anywhere in the world.

In implementing these labs it was found that the best results are achieved when students work in groups. This way group members can learn from each other and develop teamwork and communication skills. This would also satisfy *ABET Objectives 10 and 11: Communication and Teamwork.*

4.7 Curriculum Development Conclusions

A major goal of this thesis was to increase the presence of command generation in the engineering curriculum, as well as innovate new teaching methods. This goal was achieved by developing a input shaping curriculum that utilized hands-on laboratory assignments. This model was refined over several iterations in 7 classes, in 3 universities, across 3 continents. The result was two-fold; the development of crane setups that can be used by students around the world and the development of a finalized set of lab modules that teach input shaping concepts. The success of these educational tools can be seen from the high-caliber, final projects of the students that were published and presented in conferences.

4.7.1 Future Work

The educational tools developed were successful. However, future work might perform more studies to test the effectiveness of these tools. A major difficulty in evaluating education, in general, is creating a control group. One way to address this problem while evaluating the education tools developed in this thesis is to conduct a study with three engineering classes. The three classes are taught simultaneously by different professors. All three classes would receive the same input shaping lecture taught by the same professor. The first class would receive no further instruction and this class would represent the control group. The second class would use the lab sequence developed for GT Lorraine or Tokyo Tech using

a locally controlled crane. The third class would perform the final lab sequence presented in the previous section using the remotely controlled tower crane. All three classes would then be tested on input shaping concepts, as well as the core concepts: system vibration, convolution, superposition, and robustness. Comparing the results from the control group to the other two classes would indicate whether the lab sequences (local or remote) improved the students ability to learn. This study could also be used to evaluate whether there is any difference between how students learn in local and remote laboratories.

CHAPTER V

CRANE OPERATOR STUDIES

Input shaping is designed to remove the burden of minimizing payload swing while the operator drives the crane. However, it also changes the crane response to pendant commands. As a result, it will effect how the operator drives the crane.

Figure 5.1 shows the problem at hand. The major elements that move the crane include: the human operator that generates the desired response, the interface that the operator uses to control the machine, and the input shaper that transforms the desired response into a signal that controls the motors and drives. The human operator then uses visual feedback to adjust the control input. Simulations have already shown the effectiveness of input shaping and cranes. However, as Figure 5.1 shows, the operator adds a whole new dynamic to the crane-system. Therefore, an important question is: *How does input shaping effect operator performance and behavior?* It is hypothesized that input shaping makes controlling the crane easier, thereby enabling the operator to navigate the crane more efficiently. Several operator studies were conducted to test this hypothesis.

5.1 Operator Study Overview

All of the studies had the same general format. The crane operator (often a student volunteer) would be asked to drive either the HiBay, bridge, or tower crane through an obstacle course. A moderator would be present to assist them or explain how to use the crane if it was their first time. An example of one of the obstacle courses is shown in Figure 5.2. This

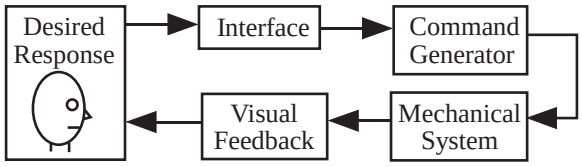


Figure 5.1: Operator-in-the-Loop Control.

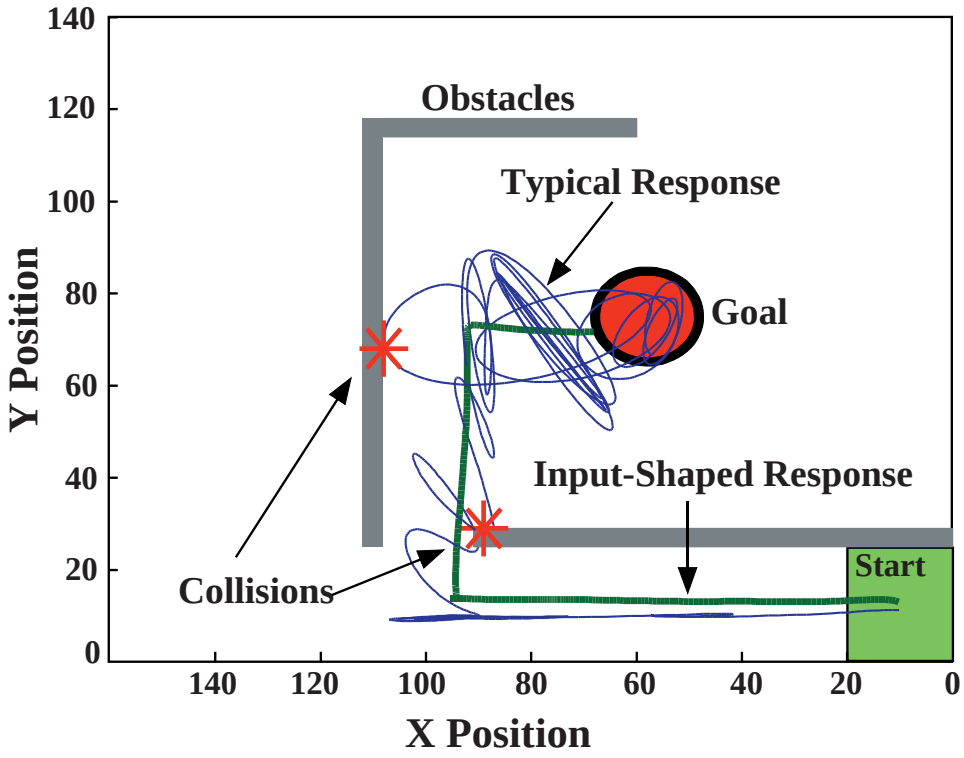


Figure 5.2: Sample Obstacle Course and Operator Response

figure shows an overhead view of the course along with the start and end points. As the operator drives through the course, the payload is tracked using an overhead digital camera. An example of payload tracking data is shown in Figure 5.2 for shaped and unshaped runs. This made it possible to record several performance measures, such as completion time, number of collisions, and what path was chosen.

The crane operator studies are summarized in Table 5.1. All the studies can be categorized as one of two modes; local or remote. In the local studies the operators were in the same vicinity as the crane they drove and had direct sight of the crane. In the remote studies some or all of the operators were driving the crane remotely over the internet using a graphical user interface (GUI). Each study measured a specific variable related to operator behavior (eg. completion time, path choice, etc...). In all cases, both shaped and unshaped trials were compared. Unshaped trials were the control since they represented "standard" crane operation. Table 5.1 also shows the section in which each study is discussed.

All of these studies took place on a large scale and required a team of many people.

Table 5.1: Summary of Operator Studies

Mode	*Name of Study*	*Variable Measured*	*Section*
Local	Completion Time	Time to reach target	5.2
	Path Choice	Type of path taken to reach target	5.3
	Number of Collisions	Number of collisions during course navigation	5.4
	Operator Learning	Performance improvement over time	5.5
Remote	Operator Performance	Completion time and collision rate	5.6
	Remote vs. Local	Local verses Remote operator performance	5.7

Those involved included: John Hued, Attir Khalid, Jacques Fortier, and Sebastien Wolff. The Tokyo Tech students included: Harald Bergur Haraldsson, Sandro Kenji Sasaki, and Eyri Watari.

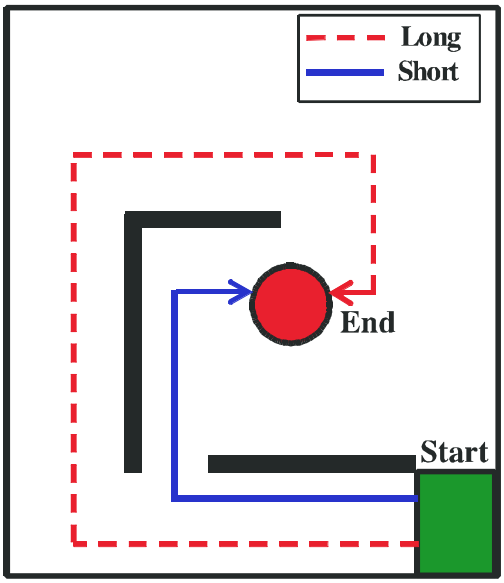
The sections that follow describe each study. In every case the objective is stated, the experimental results are presented, and the conclusions are given. The last section summarizes the conclusions from all the studies.

5.2 Local: Completion Time

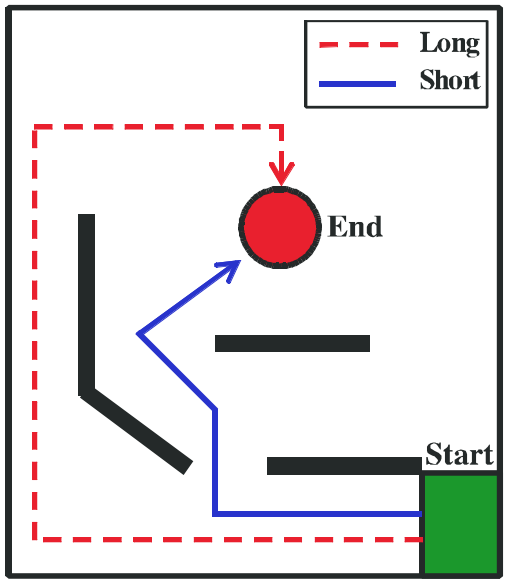
Objective: To determine the effect of input shaping on the time it takes an operator to navigate through an obstacle course. The hypothesis was that input shaping would decrease the time by eliminating payload swing

Experiments Two obstacle courses were constructed out of plywood. An overview of the obstacle courses are shown in Figures 5.3(a) and 5.3(b). Ten volunteers drove through course 1 and 13 through Course 2. For each trial the time to reach the goal was recorded. Figures 5.3(c) and 5.3(d) show the results. The time to complete the course is given for each subject.

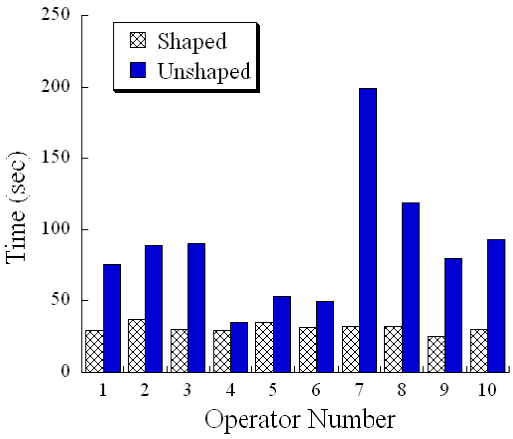
Conclusions The volunteers completed the course in a shorter amount of time when input shaping was enabled. These results are clearly shown by the data in Figures 5.3(c) and 5.3(d). The average time to complete the courses with shaping was 31 seconds and 51 seconds for Course 1 and 2 respectively. However, the average time to complete the courses without shaping was 88 seconds and 135 seconds respectively. More information on this study is reported in [33, 34].



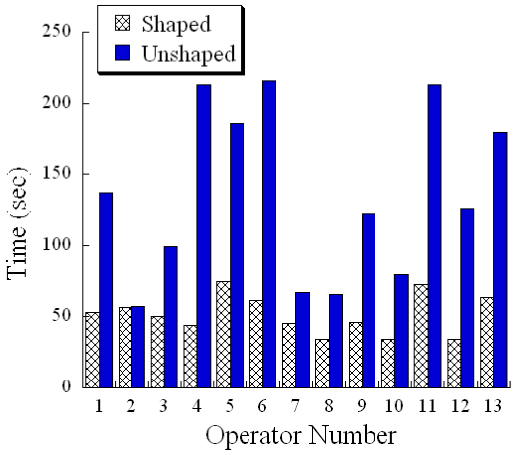
(a) Schematic Representation of Course 1.



(b) Schematic Representation of Course 2.

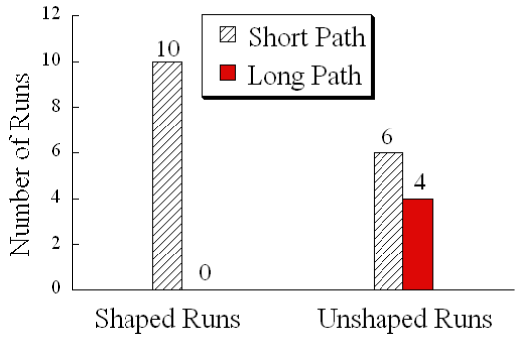


(c) Course 1 Run Times.

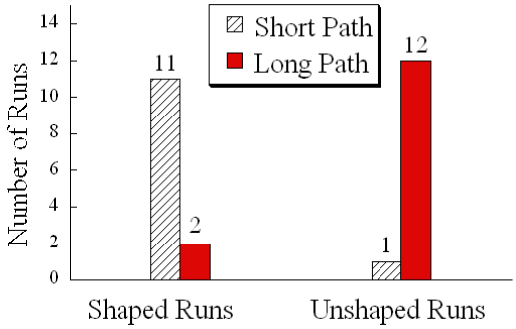


(d) Course 2 Run Times.

Figure 5.3: Overhead Sketch and Run Time Data for Courses 1 and 2.



(a) Path Choice for Course 1.



(b) Path Choice for Course 2.

Figure 5.4: Path Choice for Courses 1 and 2.

5.3 Local: Path Choice

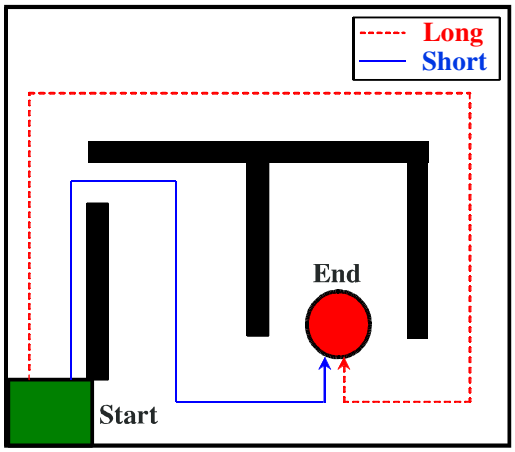
Objective: To determine the effect of input shaping on path choice through an obstacle course. The path options were: a shorter path with more obstacles, or a longer path with fewer obstacles. The hypothesis was that with input shaping operators would choose the shorter, more difficult path.

Experiments The two obstacles courses used in this study were shown in Figures 5.3(a) and 5.3(b). The two path options are also shown for each case. Notice that the shorter path is more difficult since the operator must navigate through tight space and around more obstacles. Ten volunteers drove through course 1 and 13 through course 2. For each trial the chosen path (long or short) was recorded. The results are shown in Figure 5.4(a) and 5.4(b).

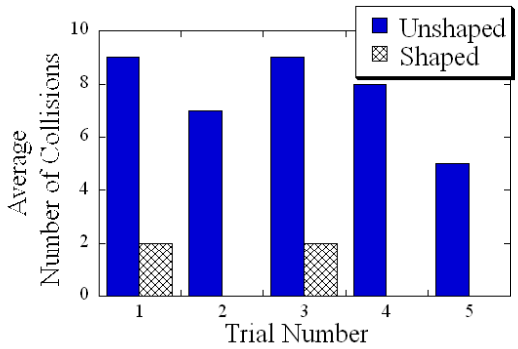
Conclusions The results clearly show that with input shaping more of the volunteers chose the shorter, more difficult path. In a real crane scenario this would be advantageous because the task would be completed in a shorter time. More information on this study is reported in [33, 34].

5.4 Local: Number of Collisions

Objective: To determine the effect of input shaping on the number of collisions as the operator navigates through an obstacle course. The hypothesis was that input shaping



(a) Overview of Course 3.



(b) Collision Data for Course 3.

Figure 5.5: Overhead Sketch and Collision Data for Course 3.

would decrease the number of collisions because the payload swings less.

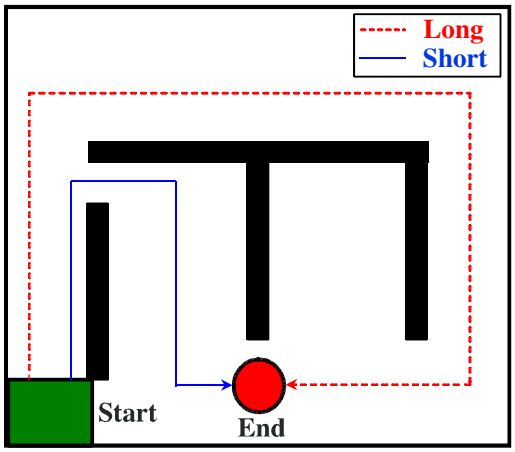
Experiments The obstacle course used for this study is shown in Figure 5.5(a). During this test a total of 12 people ran the crane between 3 and 5 times. The operators had two possible paths to take from the start to the goal. The number of collisions with the obstacles were recorded for each trial. Figure 5.5(b) shows the results. The number of collisions were averaged for each trial number across all the subjects.

Conclusions The results in Figure 5.5(b) show that input shaping caused a clear reduction in the number of collisions. More information on this study is reported in [33, 34].

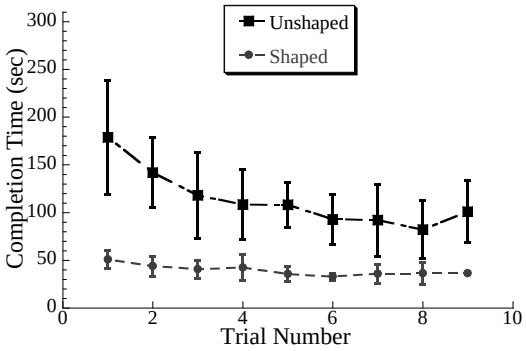
5.5 Local: Operator Learning

Objective: To determine how input shaping changes the way an operator learns to navigate through an obstacle field.

Experiments The obstacle course used for this study is shown in Figure 5.6(a) and it is a slightly modified version of the course previously shown in 5.5(a). The operators were given the task of driving from the start to the goal in a minimum time. The average testing frequency was 2.2 trials/week with each operators driving the crane between 3 and 9 times. Over the course of the study each operator's completion time was



(a) Overview of Course 4.



(b) Operator Learning Data.

Figure 5.6: Overhead Sketch and Learning Data for Course 4.

measured. This indicated how the operators learned over time. Each operator ran the crane with and without shaping. Figure 5.6(b) shows the results averaged over all the operators, as well as the standard deviation for each data point. The horizontal axis measures time in terms of trial number. The vertical axis is the time to complete the course.

Conclusions Figure 5.6(b) shows that with input shaping enabled the operators completed the course in a short amount of time, and their performance did not change significantly over time. However, without input shaping the operators initially took longer to drive the crane through the course and then learned how to drive more efficiently. So, without input shaping learning took place, but with input shaping no measurable learning occurred. Another interpretation is that input shaping makes driving the crane more intuitive and easier so there is less to learn. More information on this study is reported in [34].

5.6 Remote: Operator Performance

Objective: To determine if the same improvements in operator performance that were observed for local behavior are present in remote operator behavior. Operator performance is measured as completion time and number of collisions. The hypothesis

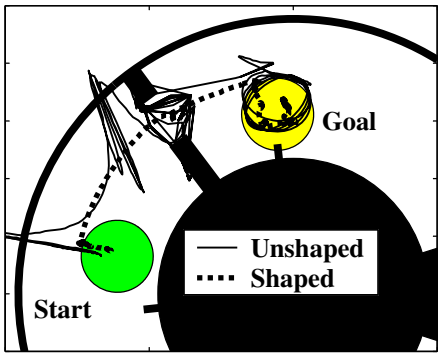


Figure 5.7: Data from Remote Operator Study.

Table 5.2: Simple Obstacle Internet Operator Study.

	Unshaped	Shaped
Avg. Completion Time (s)	63	44
Collision Rate	81%	6%

was that input shaping would yield similar improvement in operator performance for remote operation as for local operation.

Experiments The tower crane was used in this study, along with the obstacle course shown in Figure 5.7. Students in Atlanta drove the tower crane in Japan over the internet using a GUI interface, shown earlier in Figure 3.15. They were given the task of moving from the start to the goal. Eight students drove the crane a total of four times, 2 times with shaping and 2 times without. An example of a shaped and unshaped run are also shown in Figure 5.7.

The data from all of the tests is summarized in Table 5.2. The collision rate is the total number of collisions for all of the trials divided by the total number of trials. The table shows that input shaping significantly reduced the number of collisions. In addition the average completion time was reduced by 30%.

Conclusions The results showed that input shaping improved completion time and reduced the number of collisions. In other words, the improvements that were measured for remote operation were similar to those observed for local operation. More information on this study is reported in [48]

5.7 Remote vs. Local Operation

Objective: To compare remote and local operation of cranes and determine how input shaping effects the completion time and number of collisions in each case. It was hypothesized that remote operation would degrade operator performance because of communication delays and limited feedback. Furthermore, input shaping would decrease these detrimental effects because the payload is easier to control.

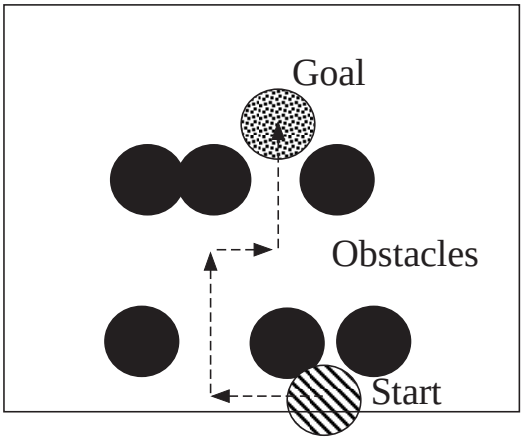
Experiments Two experimental protocols were performed. In the first experiments, students in Atlanta drove the portable bridge crane locally and remotely. The obstacle course used for these experiments is shown in Figure 5.8(a). In the second set of trials, students drove the tower crane in Japan locally and remotely. The obstacle course used for these experiments is shown in Figure 5.8(b). For all trials, the completion time and number of collisions were recorded. The completion time for the two sets of trials are shown in Figures 5.8(c) and 5.8(d) while the number of collisions are shown in Figures 5.8(e) and 5.8(f).

Conclusions The results show that remote operation caused more collisions and longer completion times for the unshaped case. However, with input shaping enabled, the operators performed better. Furthermore, with input shaping, switching to remote operation did not cause as much degradation in performance. More information on this study is reported it [30].

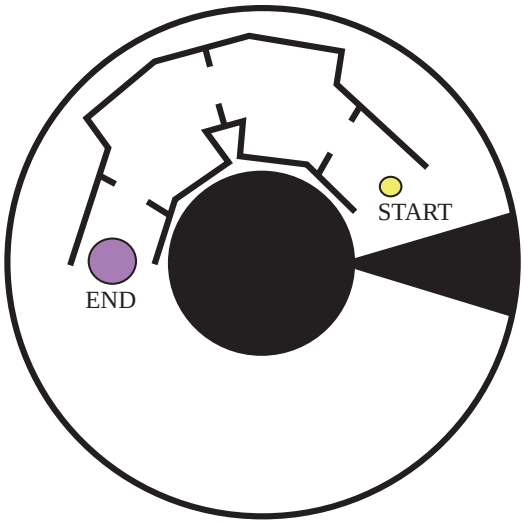
5.8 Operator Study Conclusions

The results from all of the operator studies are summarized in Table 5.3. The table lists each study and summarizes the conclusions for shaped and unshaped operation. The performance of shaped and unshaped operation is also compared for each study. The table shows that in every scenario input shaping yielded superior performance.

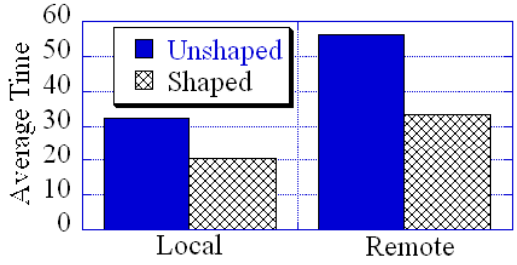
The original question which motivated these studies was: *How does input shaping effect operator performance?*. Looking at Figure 5.3 it is clear that input shaping improves operator performance over a wide range of variables, operating conditions, and scenarios.



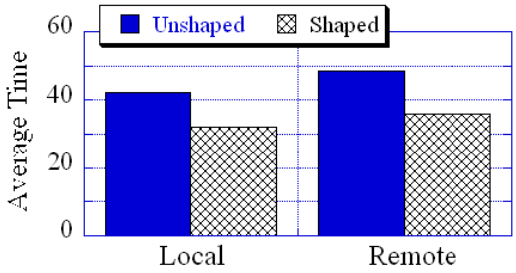
(a) Overview of Course 5.



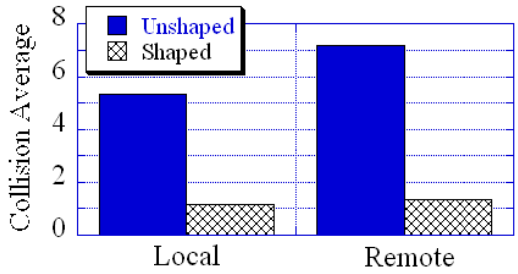
(b) Overview of Course 6.



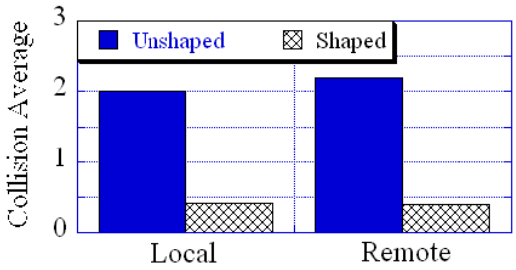
(c) Course 5 Run Times.



(d) Course 6 Run Times.



(e) Course 5 Collisions.



(f) Course 6 Collisions.

Figure 5.8: Overhead Sketch, Run-Time Data, and Collision Data for Courses 5 and 6.

Table 5.3: Crane Operator Summary
: Indicates Superior Performance

		Conclusions	
Mode	**Study Name**	**Shaped**	**Unshaped**
Local	Completion Time	Shorter	Longer
	Path Choice	Shorter (more aggressive)	Longer (easier)
	Number of Collisions	Less	More
	Operator Learning	No Learning Needed (intuitive)	Slower Learning Curve
Remote	Operator Performance	Shorter Time Less Collisions	Longer Time More Collisions
	Local vs. Remote	Minimal Effect from Remote Operation	Substantial Degradation from Remote Operation

CHAPTER VI

COMMAND SHAPING FOR A FLYBALL SYSTEM

The nonlinear swing dynamics of tower cranes pose a significant challenge to command shaping control when the slew rate is high or when the trolley is far out on the jib. In order to develop a basic understanding of this nonlinearity, a simpler flyball system is studied first. The flyball system is shown in Figure 6.1. The system consists of a central rotating mast that has position $\theta(t)$ and angular velocity $\sigma(t)$. At the top of the mast is an arm with length R, affixed perpendicular to the mast. At the end of the arm a linkage is attached via a pivot. The linkage is of length L, and can only rotate in the plane formed by the mast and arm. At the end of the linkage is a mass. The linkage is assumed to be massless.

As the mast rotates, the mass at the end of the linkage will swing outward due to centrifugal forces. This causes a deflection angle ϕ. As an example, consider driving the system with a pulse function for σ, as shown by the solid line in Figure 6.2(a). The pulse has a amplitude of σ_f, and a duration of, t_p. On a tower crane, the pulse duration would represent how long the operator holds one of the directional velocity buttons on the pendant. The resulting deflection of the mass, ϕ, is shown in Figure 6.2(b) as the solid line. The motion clearly causes a vibratory response. Notice that while the mast is rotating, the mass

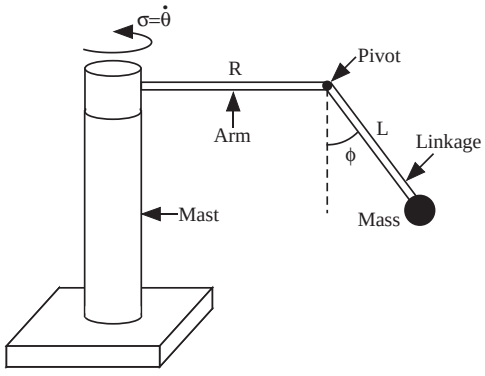
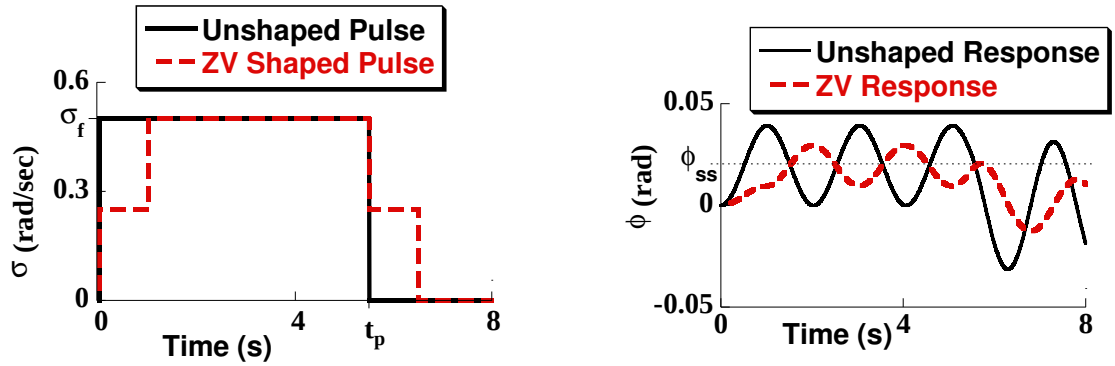


Figure 6.1: One Mass Flyball System.



(a) Pulse Velocity Command.

(b) Pulse Velocity Response.

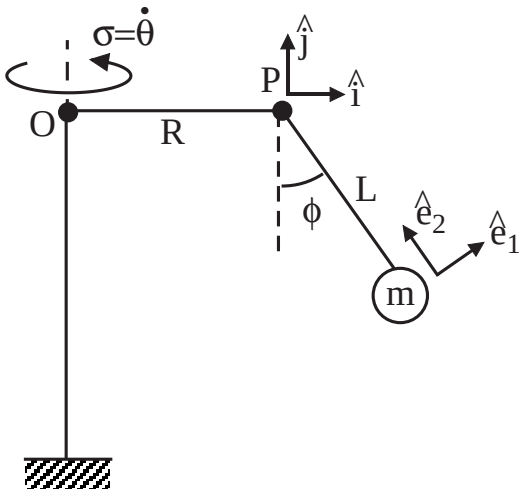
Figure 6.2: Response of the Flyball System to ZV shaped and Unshaped Commands.

oscillates about a non-zero equilibrium value. When it comes to a stop at the end of the velocity pulse, the linkage mass then oscillates about the $\phi = 0$ location.

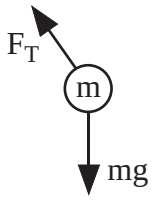
The goal of this section is: *To create an new input shaped command that will rotate the flyball system without causing residual oscillation.* As mentioned before, the flyball system is very similar to a rotating tower crane. However, the mass is restricted to only swing radially, in the plane of the arm and mast. The hope is that deriving input shapers for this simplified system will lead to insight for designing shapers for tower cranes.

Suppose a standard ZV shaped command is given to the system, as shown by the dashed line in Figure 6.2(a). The ϕ-response is shown by the dashed line in Figure 6.2(b). While the ZV shaper has decreased the vibration by 60%, its performance is far from ideal. The ZV shaper's ineffectiveness is due to the non-linear dynamics of the rotating system. This chapter seeks to find new command shaping techniques that compensate for this non-linearity.

The rest of this chapter is devoted to solving the input shaping problem for the flyball system. A model of the system is developed in the next section 6.1. Then, three new types of shapers are derived in Sections 6.2, 6.3, and 6.4 using different solution strategies. A unity magnitude shaper for the system is derived in Section 6.5. The conclusions drawn from this analysis are presented in Section 6.6.



(a) Detail Sketch of Flyball System for Dynamic Analysis.



(b) Freebody Diagram of Mass for Flyball System.

Figure 6.3: Sketch and Freebody Diagram of Flyball System.

6.1 Dynamic Analysis of the Flyball System

The equations of motion for the 1-mass flyball system can be derived using standard Newton-Euler analysis. A detailed sketch of the system is given in Figure 6.3(a). The $\hat{\imath}, \hat{\jmath}, \hat{k}$ coordinate system is fixed to the pivot point P, which is rotating around point O in a horizontal plane. The $\hat{e_1}, \hat{e_2}, \hat{e_3}$ coordinate system is fixed to the mass, m. The $\hat{k}$ and $\hat{e}_3$ vectors are both directed out of the page.

A free body diagram of the mass is shown in Figure 6.3(b). Force $\vec{F_T}$ represents the tension force exerted on the mass by the linkage. The Newton-Euler force balance for this system is:

$$\begin{aligned} \sum \vec{F} &= m\vec{a}_m \\ F_T \cdot \hat{e_2} - mg \cdot \hat{\jmath} &= m\vec{a}_m \end{aligned} \tag{6.1}$$

where $\vec{a}_m$ is the acceleration of mass m.

The acceleration of mass m can be derived by using $\hat{\imath}, \hat{\jmath}, \hat{k}$ as a intermediate coordinate system. The equation that relates the acceleration of m to the moving coordinate system $\hat{\imath}, \hat{\jmath}, \hat{k}$ is [22]:

$$a_m = \vec{a}_P + \vec{a}_{m/P} + 2\vec{\omega}_P \times \vec{v}_{m/P} + \vec{\alpha}_P \times \vec{r}_{m/P} + \vec{\omega}_P \times \vec{\omega}_P \times \vec{r}_{m/P} \tag{6.2}$$

where,

$$\begin{aligned}
\vec{a}_P &= \text{Acceleration of P} \\
\vec{\omega}_P &= \text{Angular Velocity of P-frame}(\hat{\imath}, \hat{\jmath}, \hat{k}) \\
\vec{\alpha}_P &= \text{Angular Acceleration of P-frame}(\hat{\imath}, \hat{\jmath}, \hat{k}) \\
\vec{r}_{m/P} &= \text{Position vector from P to m} \\
\vec{v}_{m/P} &= \text{Velocity vector of m relative to P-frame}(\hat{\imath}, \hat{\jmath}, \hat{k}) \\
\vec{a}_{m/P} &= \text{Acceleration of m relative to P}
\end{aligned}$$

Using Figure 6.3(a), each term of the acceleration equation can be derived:

$$\begin{aligned}
\vec{a}_P &= -R\sigma^2 \cdot \hat{\imath} - R\dot{\sigma} \cdot \hat{k} \\
\vec{a}_{m/P} &= L\dot{\phi}^2 \cdot \hat{e_2} + L\ddot{\phi} \cdot \hat{e_1} \\
2\vec{\omega}_P \times \vec{v}_{m/P} &= -2L\sigma\dot{\phi}\cos(\phi) \cdot \hat{e_3} \\
\vec{\alpha}_P \times \vec{r}_{m/P} &= -L\sin(\phi)\dot{\sigma} \cdot \hat{k} \\
\vec{\omega}_P \times \vec{\omega}_P \times \vec{r}_{m/P} &= -L\sin(\phi)\sigma^2 \cdot \hat{\imath}
\end{aligned} \tag{6.3}$$

All of these terms can be rewritten using the $\hat{e_1}, \hat{e_2}, \hat{e_3}$ coordinate system by applying the coordinate transformation: $\hat{\jmath} = \cos\phi\hat{e_2} + \sin(\phi)\hat{e_1}$, $\hat{\imath} = -\sin\phi\hat{e_2} + \cos\phi\hat{e_1}$, $\hat{k} = \hat{e_3}$. Then, the terms in (6.3) can be substituted back into (6.2) to yield the total acceleration of m. This, in turn, can be substituted into (6.1) to yield the equations of motion for the system. Because the mass is constrained to move in the $\hat{e_1}$ direction, only terms that contain this vector are relevant to the equations of motion. All other terms can be eliminated. The resulting equation of motion for the system is:

$$\ddot{\phi} + \frac{g}{L}\sin(\phi) = \frac{R + L\sin(\phi)}{L}\cos(\phi) \cdot \sigma^2 \tag{6.4}$$

The left hand side of (6.4) is identical to the homogeneous terms of a undamped-pendulum. The right hand side of (6.4) is the forcing function and the input is the angular velocity σ. This right-hand side comes from the centripetal acceleration of mass m. It originates from the acceleration terms $\vec{a}_{m/P}$ and $\vec{\omega}_P \times \vec{\omega}_P \times \vec{r}_{m/P}$ that appeared in the acceleration equation (6.2). Similar centripetal effects will show up in the tower crane analysis. This is why studying the flyball system is helpful toward understanding tower cranes.

6.1.1 Equilibrium at Constant Angular Velocity

The system response to an angular velocity pulse was shown in Figure 6.2(b). Recall that while the mast is rotating, the mass oscillates about a non-zero equilibrium angle, ϕ_{ss}. This steady-state angle will play an a important role in deriving effective command shapers. The steady-state angle can be derived by setting the ϕ-acceleration term in the equations of motion equal to zero:

$$0 = x_1 \cos(\phi_{ss}) + x_2 \sin(\phi_{ss}) \cos(\phi_{ss}) - \sin(\phi_{ss}) \tag{6.5}$$

$$x_1 = \left(\frac{R\sigma_{ss}^2}{g} \right) \qquad\qquad x_2 = \left(\frac{L\sigma_{ss}^2}{g} \right)$$

where ϕ_{ss} is the steady-state deflection angle, and σ_{ss} is the steady state angular velocity of the mast. In the example shown in Figure 6.2(b), the steady state velocity is given by $\sigma_{ss} = \sigma_f$. Notice that even though there are three input parameters(ϕ_{ss}, L, R), the steady-state deflection is a function of only two non-dimensionalized constants (x_1, x_2). Both of these constants represent the ratio between a centripetal acceleration and the acceleration due to gravity. The centripetal acceleration of point P, in Figure 6.1, is contained in the numerator of x_1. The centripetal acceleration of m relative to P is given by $L\sigma^2 sin\phi$, which forms the basis for x_2. Note that the total centripetal acceleration of the mass m is the sum of these two accelerations.

A simplified equation for the steady-state deflection can be derived using a small angle approximation. If we assume that $\phi_{ss} < 0.3$ rad, then the sine and cosine functions can be approximated as: $\sin(x) \approx x$, $\cos(x) \approx 1$. Applying this to (6.5) yields the linearized equations:

$$\phi_{ss} = \frac{x_1}{1 - x_2} \tag{6.6}$$

$$x_1 = \left(\frac{R\sigma_{ss}^2}{g} \right) \qquad\qquad x_2 = \left(\frac{L\sigma_{ss}^2}{g} \right)$$

Figure 6.4 shows ϕ_{ss} for various values of x_1 and x_2. The solid lines are the numerical solutions for ϕ_{ss} using (6.5). The dotted lines are the linear approximations for ϕ_{ss} using (6.6). Notice that the linearized solution gives a good approximation for $\phi < 0.3$, as stated earlier. Also notice that the linearized solution is undefined for $x_2 \geq 1$.

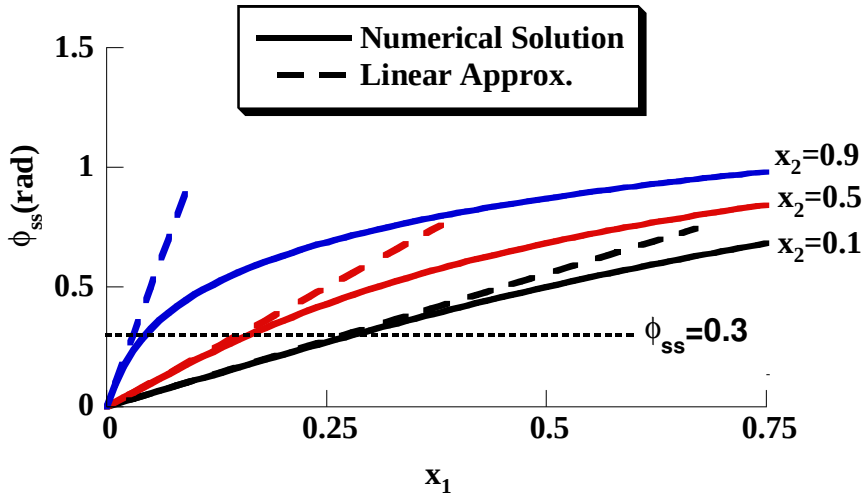


Figure 6.4: Steady-State Deflection ϕ_{ss} for various parameters.

6.2 A ZV Shaper for a Partially Linearized Flyball Model

A simple ZV shaper can be derived for the 1-mass flyball system by partially-linearizing its equations of motion. Two assumptions are utilized for the linearization:

Assumption #1: A small angle approximation can be used for ϕ yielding: $\sin(\phi) \approx \phi$, $\cos(\phi) \approx 1$.

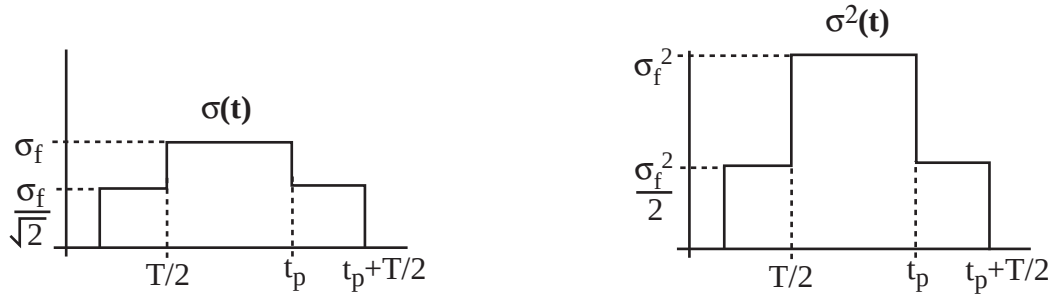
Assumption #2: The term $\frac{R+L\sin(\phi)}{L}$ appearing in the right-hand side of the equations of motion can be approximated as $\frac{R}{L}$.

The set of system parameters that satisfy these assumptions will be discussed later. Using these two assumptions the equations of motion can be rewritten as:

$$\ddot{\phi} + \frac{g}{L}\phi = \frac{R}{L} \cdot \sigma^2 \tag{6.7}$$

Notice that (6.7) is only partially linearized due to the presence of the σ^2 term.

If the σ^2 term is ignored, (6.7) is a linear, second-order oscillator with a natural frequency of $\sqrt{g/L}$. The effect of the σ^2 term is only to warp the amplitude of the input signal. However, the timing of the input signal is unaffected. With this in mind, a zero-vibration command can be formed by taking a standard ZV shaped pulse and taking the square-root



(a) Original $\sigma(t)$ Command.

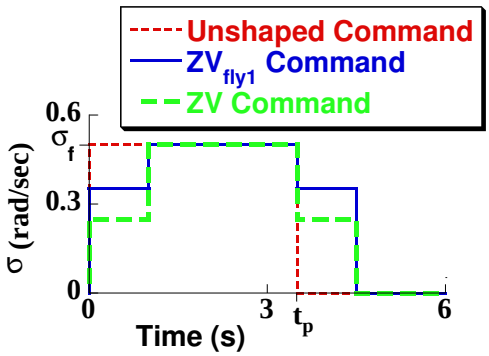
(b) $\sigma(t)^2$ Driving Function.

Figure 6.5: Forming a ZV Command for the Linearized Flyball Model

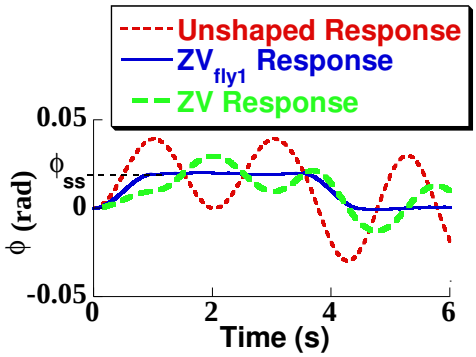
of the values. When this signal is used to command the system the squaring effect will reconstruct the ZV shaped pulse, which will move the linear left-hand-side of (6.7) with zero vibration. This zero-vibration command for a flyball system, termed ZV_{fly1}, is shown in Figure 6.5(a). T is the linearized period of the system given by $\frac{2\pi}{\sqrt{g/L}}$, t_p is the pulse time or desired command length, and σ_f is the maximum or setpoint angular velocity. Figure 6.5(b) shows the command with the amplitude squared. This is the forcing function that drives the linearized equations of motion in (6.7).

Figure 6.5(b) is identical to an input-shaped pulse command. It is tuned for the natural frequency of the linearized model (6.7). Therefore, it will induce zero residual vibration. This proves that the command shown in Figure 6.5(b) is a ZV solution for the system with the σ^2 term acting on the input. This solution will be called ZV_{fly1}. Notice that the "trick" to forming this command is to modify the intermediate step of the $\sigma(t)$ command. In a normal ZV shaped pulse, this intermediate step would have amplitude of $\sigma_f/2$, whereas this command uses an amplitude $\sigma_f/\sqrt{2}$ in anticipation of the squaring effect in (6.7).

Figures 6.6(a) and 6.6(b) compare the ZV_{fly1} shaped, ZV shaped, and unshaped command and response. All of these responses were formed using the actual equations of motion for the flyball system (6.4), not the linearized form. Notice that the ZV_{fly1} command has significantly less vibration than both the unshaped and ZV responses.

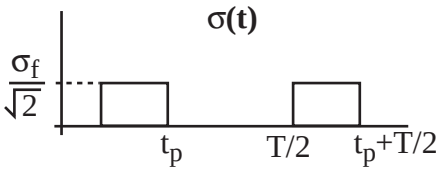


(a) ZV_{fly1}, ZV, and Unshaped Commands.

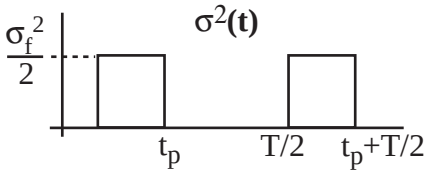


(b) ZV_{fly1}, ZV, and Unshaped Response.

Figure 6.6: ZV_{fly1}, ZV, and Unshaped Simulations [L=1 m, R=0.75 m, $\sigma_f = 0.5$ m/s].



(a) Original $\sigma(t)$ Command.



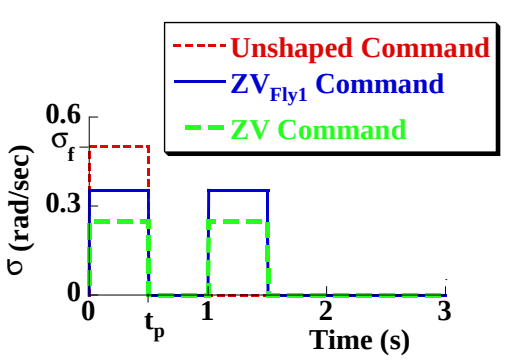
(b) $\sigma(t)^2$ Driving Function.

Figure 6.7: Forming a ZV Short Command for the Linearized Flyball Model.

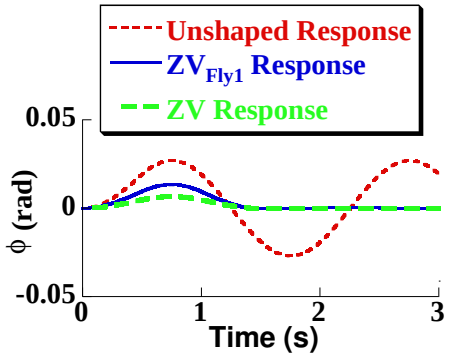
6.2.1 ZV_{fly1} Shaped Short Commands

The ZV_{fly1} solution illustrated in Figure 6.5 is incomplete. It will not work effectively if the pulse time is less than one-half the system period: $t_p < T/2$. Under these conditions a standard ZV shaped pulse takes the form of two short pulses, rather than the single pulse with a two-step staircase shown in Figure 6.5. However, the same strategy can be used to find a flyball command for these two pulses: take the square root of the values to compensate for the squaring effect in the dynamics. The ZV_{fly1}command for short command durations is shown in Figure 6.7. On the left, Figure 6.7(a) shows the σ command. On the right, Figure 6.7(b) shows the command with its amplitude squared. The squared command is an input shaped command; therefore, the linearized model (6.7) will move with zero residual vibration. Similar to the previous case, this solution is found by anticipating the effect of the squaring and compensating for it in the σ command.

To differentiate between the two solutions, Figure 6.5 will be referred to as a ZV_{fly1} *long command*. It is valid for $t_p > T/2$. Figure 6.7 will be referred to as a ZV_{fly1} *short command*

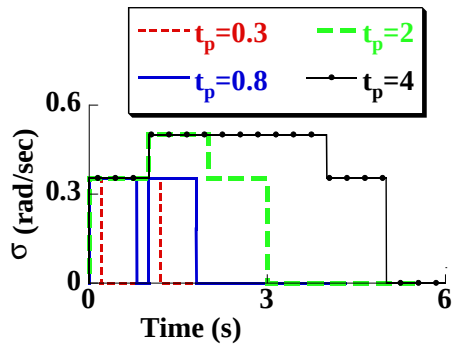


(a) ZV_{fly1}, ZV, and Unshaped Commands.

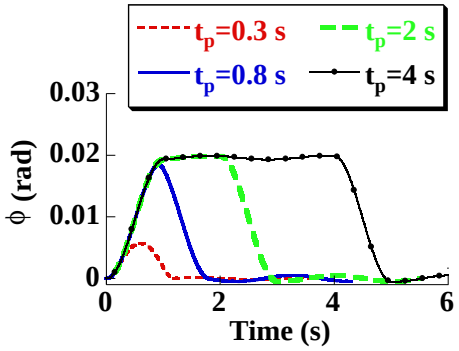


(b) ZV_{fly1}, ZV, and Unshaped Response.

Figure 6.8: ZV_{fly1}, ZV, and Unshaped Simulations [L=1 m, R=0.75 m, $\sigma_f = 0.5$ m/s].



(a) ZV_{fly1}Commands.



(b) ZV_{fly1}Responses.

Figure 6.9: ZV_{fly1}Commands and Responses for Various t_p.

which is valid for $t_p < T/2$.

Figure 6.8 compares ZV_{fly1}, ZV, and unshaped short commands. Figure 6.8(a) shows the commands and Figure 6.8(b) shows the response. The ZV_{fly1} command eliminates the residual vibration. Notice that the ZV short command also eliminates the vibration. Recall from Figure 6.7 that the only difference between the standard ZV command, and the ZV_{fly1} command is a scaling factor. Since the residual vibration is zero this scaling factor has no effect on the residual vibration.

The *short* and *long* ZV_{fly1} algorithms can be combined into a single algorithm. Figure 6.9(a) shows what the ZV_{fly1} commands would look like using various t_p values. Figure 6.9(b) shows what the corresponding ZV_{fly1} responses would be.

6.2.2 Discussion of $\mathbf{ZV_{fly1}}$ Assumptions

The ZV_{fly1} solution is based on the linearized flyball model (6.7). As mentioned earlier this model is based on two assumptions:

Assumption #1 A small angle approximation can be used for ϕ yielding: $\sin(\phi) \approx \phi$, $\cos(\phi) \approx 1$.

Assumption #2 The term $\frac{R+L\sin(\phi)}{L}$ appearing in the right-hand side of the equations of motion can be approximated as $\frac{R}{L}$.

These assumptions restrict the parameter space over which the ZV_{fly1} shaper, and the model it was developed from, are valid. This section will examine this range.

The first assumption limits the maximum deflection angle of the response. Figure 6.6(b) showed an example of a ZV_{fly1} response. Notice that the maximum deflection angle equals the steady-state deflection, ϕ_{ss} for the setpoint speed σ_f. This is a necessity because if $\phi \neq \phi_{ss}$ then $\ddot{\phi}$ would be non-zero in the equations of motion (6.4). This would imply a non-zero vibration amplitude which is not the case for a shaped response.

Assumption #1 will be valid when the maximum deflection angle is less than about 0.3 rad. Based on the argument in the preceding paragraph, this is equivalent to constraining the steady-state deflection to be less than 0.3 rad:

$$\max(\phi) = \phi_{ss} < 0.3rad \tag{6.8}$$

The steady state deflection can be numerically derived from (6.5) and was plotted in Figure 6.4. However, notice that (6.8) implicitly assumes that $\phi_{ss} < 0.3$. Earlier, it was shown that under these conditions a linearized approximation (6.6) for ϕ_{ss} could be used. Substituting this approximation into (6.8) yields a sufficient condition for assumption #1 to hold:

$$x_1 \leq 0.3 - 0.3x_2 \tag{6.9}$$

where x_1 and x_2 are the non-dimensionalized parameters defined in (6.6).

The second assumption can be rewritten as:

$$\frac{R}{L} \gg \max(\sin(\phi)) \tag{6.10}$$

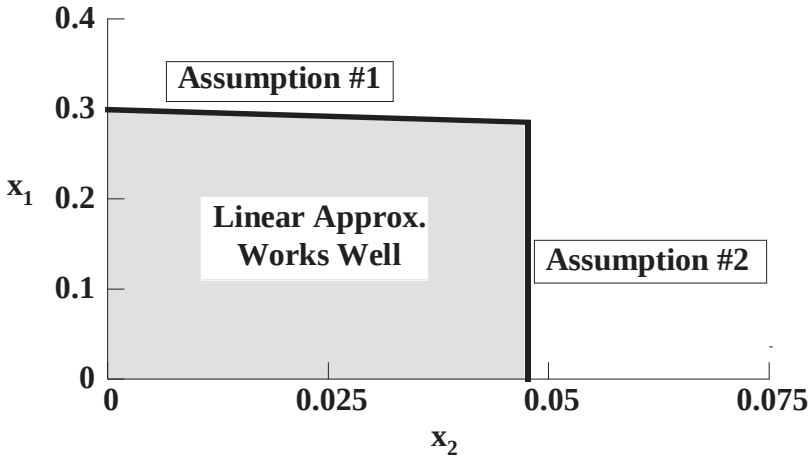


Figure 6.10: Parameter Space in which Linearizing Assumptions are Valid

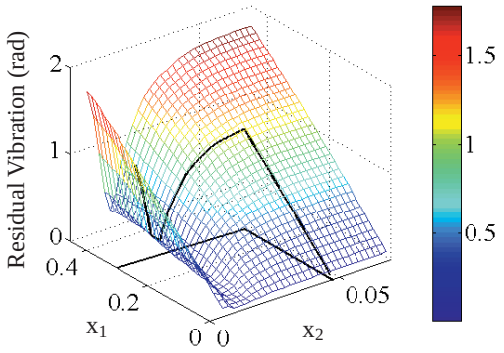
Assuming that assumption #1 holds, the $\sin(\phi)$ term can be replaced with ϕ. It will be assumed that so long as $\max(\sin(\phi))$ is 5% or less of $\frac{R}{L}$, the relationship will satisfied. Using these ideas, (6.10) can be rewritten as:

$$x_2 < 0.048 \tag{6.11}$$

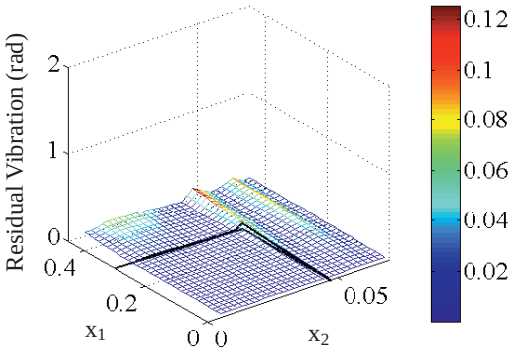
where x_2 is the non-dimensionalized parameter defined in (6.6).

Figure 6.10 shows a plot of the operational space where the two assumptions hold. The horizontal axis is x_1 and the vertical axis is x_2. The grey area shows the region where (6.9) and (6.11) are satisfied and the assumptions hold. This is the region where the ZV_{fly1} shaper will perform well. The top border of the region represents assumption #1 and the right border represents assumption #2.

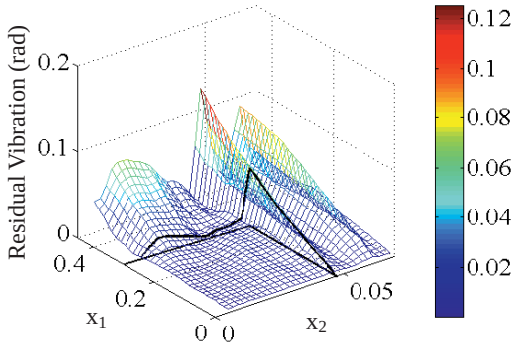
Figure 6.10 gives a qualitative feel for how the system parameters effects the non-linear behavior of the system. Both x_1 and x_2 have a factor of σ_f^2, therefore according to the figure increasing the speed makes the system more non-linear. Increasing L will increase x_2. According to the figure this makes the system more nonlinear predominantly by violating assumption #2. Increasing R will increase x_1. According to the figure this makes the system more nonlinear predominantly by violating assumption #1.



(a) Unshaped Residual Vibration for $\sigma_f = 0.25$ m/s.



(b) ZV_{fly1} Residual Vibration for $\sigma_f = 0.25$ m/s.



(c) Close-up View of Figure 6.11(b).

Figure 6.11: Comparison of Residual Vibration of Unshaped and ZV_{fly1} Shaped Commands.

6.2.3 ZV_{fly1} Performance Over a Range of Parameters

The residual vibration when using ZV_{fly1} shaped and unshaped commands on the flyball system were compared over a range of parameters. Figure 6.11(a) shows the unshaped vibration and Figure 6.11(b) shows the ZV_{fly1} performance. The linkage length L, and arm length R were both varied, while the final velocity σ_f was held constant. The parameter space is shown in terms of the non-dimensionalized values x_1 and x_2. The vertical axis gives the residual vibration in radians. The black lines give the boundaries where assumptions #1 and #2 are violated. This boundary is drawn on the horizontal plane, and also projected onto the 3-D surface for easy viewing.

Comparing figures 6.11(a) and 6.11(b), it is clear that the ZV_{fly1} shaped command

outperforms the unshaped command over the entire parameter space. Note that both surfaces are plotted using the same axes limits. Figure 6.11(c) shows a close-up view of the ZV_{fly1} data. This plot shows that outside the region bounded by the black lines, the shaper performance degrades. Inside these boundaries the ZV_{fly1} shaper performs very well. This verifies that (6.9) and (6.11) accurately predict the parameter space where the ZV_{fly1} shaper is valid. However, even outside of this range, the ZV_{fly1}shaper significantly reduces vibration when compared to the unshaped response.

6.3 A Numerically Derived ZV Shaper for the Full Flyball Model

The last section showed that the ZV_{fly1} command worked best in a limited parameter space. This section will develop a numerical approach for finding a ZV command that does not have the same limitations. The strategy for this approach comes from an earlier observation about the ZV response: The flyball deflection, ϕ, is equal to the steady-state deflection, ϕ_{ss}, while the mast spins at constant velocity σ_f. Therefore, the command should be chosen to achieve this steady-state condition. A similar technique was used by Smith, et. al. in [92] for a system with a linearly varying natural frequency. The approach here is to use a "shooting" technique to find a two-step command that achieves this condition. It is called shooting because the command is adjusted to make the simulated response hit the targeted final condition at the right time. This is a common strategy for solving boundary value problems.

A detailed representation of the numerical algorithm appears in Figure 6.12. To distinguish this command from the previous one, this numerical approach will be called ZV_{fly2}. The inputs to the algorithm are the system parameters: R, L, and σ_f, the final setpoint velocity. The first step is to predict the steady state deflection ϕ_{ss}. This is done by numerically solving (6.5).

The second step is to find the amplitude of a step, whose ϕ-response peaks at ϕ_{ss}. The right hand side of Figure 6.12 details how this works. Again, the strategy is a "shooting"

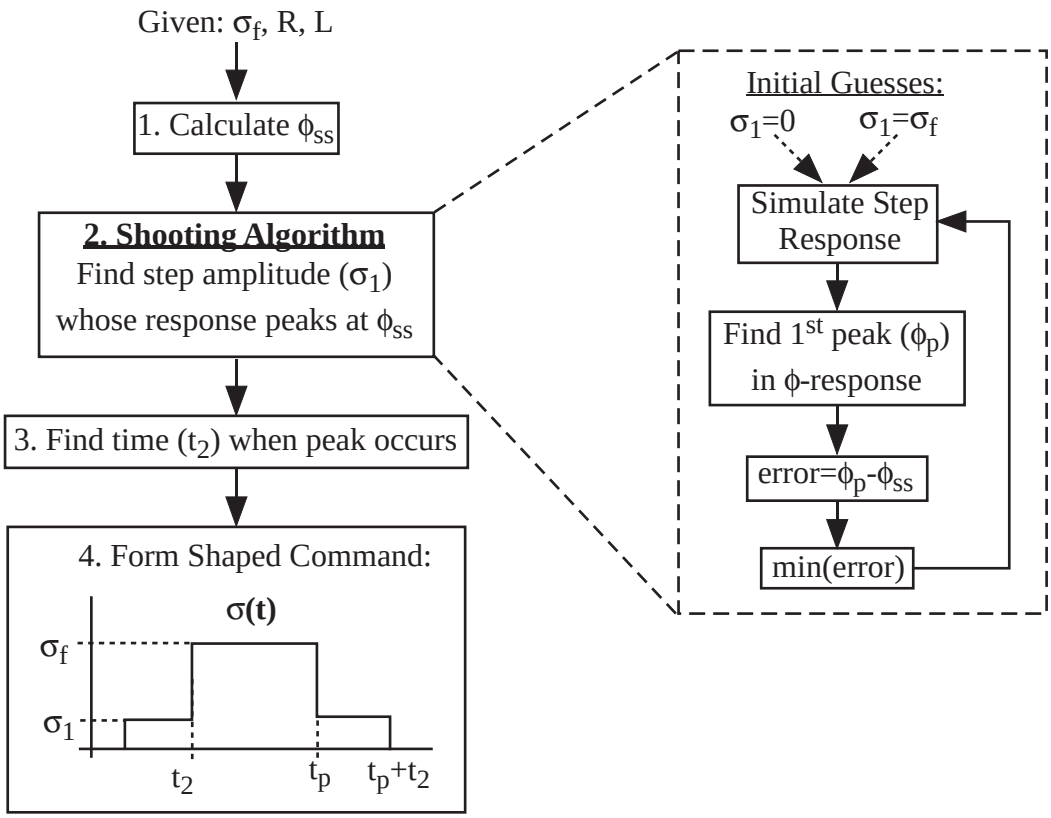
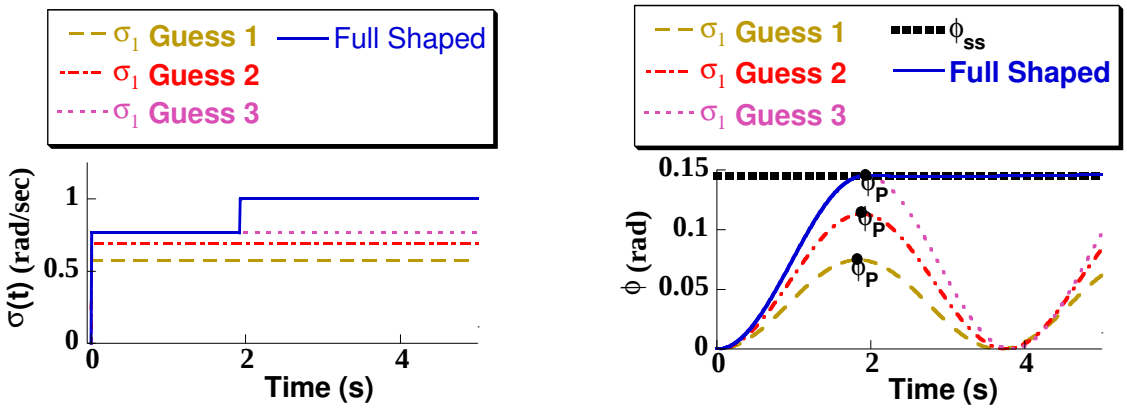


Figure 6.12: Outline of ZV_{fly2} Algorithm.



(a) "Shooting" Algorithm Guesses.

(b) "Shooting" Algorithm Responses.

Figure 6.13: "Shooting" Algorithm Command Guesses and Responses.

method, similar to the one used to solve boundary-value problems. The system is simulated using a guess for the step command, labeled as "Guess 1" in Figure 6.13(a). The corresponding ϕ-response is shown in Figure 6.13(b). The error is measured as the distance between the first peak of the simulated response and the steady state deflection, ϕ_{ss}. A numerical solver is then used to minimize the error. In this case, the Matlab function *fzero* was used. The initial guesses $\sigma_1 = 0$ and $\sigma_1 = \sigma_f$ were used since the answer is guaranteed to lie between these two values. This bounded the solution space and led to faster convergence than using an unbounded solution space. Figures 6.13(a) and 6.13(b) show how the optimization process might progress. In this example the process converges on the third guess.

The third step is to measure the time, t_2, at which the peak, ϕ_P, occurs. In the example shown in Figure 6.13(b) this time is approximately 2 sec.

The fourth, and final step in the process is to form the shaped command. Values σ_1 and t_2 define the amplitude and duration of the intermediate step, as shown in the bottom of Figure 6.12. Figures 6.13(a) and 6.13(b) also show the final shaped command and response for this example. The first step of the command makes the deflection peak at ϕ_{ss}. At that exact instant, t_2, the velocity command switches to the final setpoint velocity, σ_f. Since the deflection has a zero-derivative at this point, and since it is at the steady-state value for σ_f there will be no residual oscillation about ϕ_{ss}.

The above strategy only finds the "rising" portion of the shaped command that brings the system to a steady-state angular velocity. To find the "falling" portion of the shaped command that returns the system to rest, the above strategy could be repeated. However, a short-cut can be used instead. It is hypothesized that the "falling" portion of the command is really the mirror image of the "rising" portion, as shown at the bottom of Figure 6.12. The following argument proves this hypothesis:

1. The "rising" portion of the command solves a boundary value problem (BVP). It brings the system described by $\ddot{\phi} + \frac{g}{L}\sin(\phi) = \frac{R+L\sin(\phi)}{L}\cos(\phi)\cdot\sigma^2$ from the initial state $[\phi(t_1) = 0, \sigma(t_1) = 0]$ to the final state $[\phi(t_2) = 0, \sigma(t_2) = \sigma_f]$.

2. The "falling" portion of the command needs to solve a similar BVP. It must bring the system described by from the initial state $[\phi(t_1) = 0, \sigma(t_1) = \sigma_f]$ to the final state $[\phi(t_2) = 0, \sigma(t_2) = 0]]$.

3. If time progresses in the negative direction, then the second BVP becomes equivalent to the first BVP. Note that the equations of motion are the same whether dt is positive or negative.

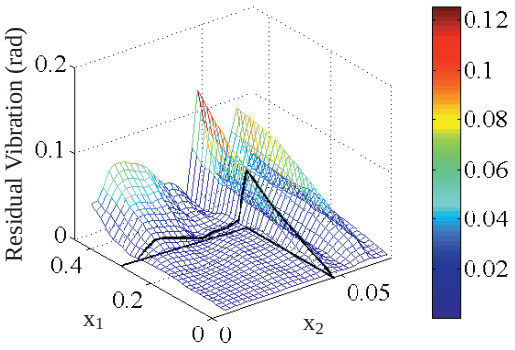
4. Since the two BVPs are equivalent, they have equivalent solutions. Therefore, the "falling" portion of the command must be the mirror image of the "rising" portion. In other words, the command is symmetric in velocity about its mid-point, but anti-symmetric in acceleration.

Note that the ZV_{fly1} commands derived earlier were symmetric as well.

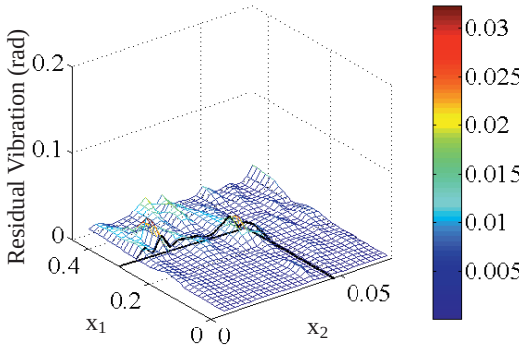
The ZV_{fly2} solution developed above only applies for sufficiently long values of t_p, the pulse time. As was the case for the ZV_{fly1} command, short values of t_p result in shaped commands with entirely different shapes. This can be seen comparing the long command shown in Figure 6.5 with the short command shown in Figure 6.7. Since the above process was developed with a *long command* template, it does not apply to *short commands*.

6.3.1 Comparison of ZV_{fly1} and ZV_{fly2} Commands

The performance ZV_{fly1} and ZV_{fly2} commands are compared in Figures 6.14(a) and 6.14(b). Figure 6.14(a) is identical to Figure 6.11(c) and is repeated for easy comparison. As in previous plots, the vertical axis measures residual vibration and the horizontal plane is the parameter space. Comparing the two plots, the ZV_{fly2} shaper has significantly lower residual vibration. Notice that the ZV_{fly2} shaper works well both inside and outside the "linear" region defined by the black lines. In theory, the ZV_{fly2} shaper should have exactly zero vibration for all values. Any non-zero measurements are the result of numerical error in the ZV_{fly2} algorithm. Figure 6.15 demonstrates this fact. Figure 6.15(a) shows the ϕ-response using two different simulation resolutions: 0.01 sec and 0.001 sec. Figure 6.15(b) shows a close up of the residual vibration for this simulation. Notice that with the higher

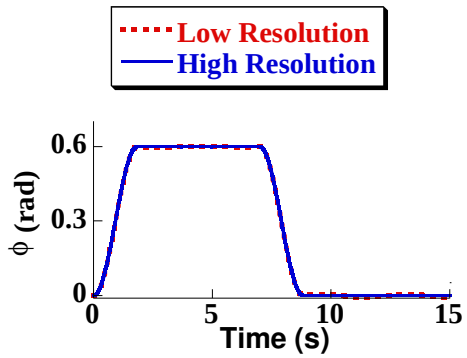


(a) ZV_{fly1} Residual Vibration.

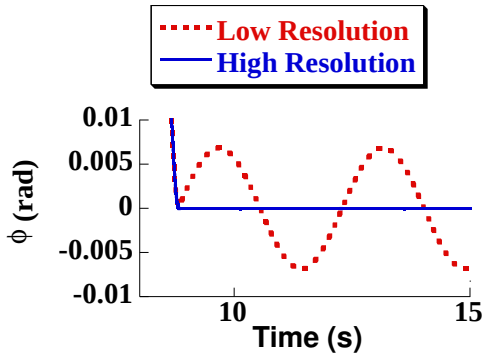


(b) ZV_{fly2} Residual Vibration.

Figure 6.14: ZV_{fly1} and ZV_{fly2} Residual Vibration for $\sigma_f = 0.25$ m/s.



(a) ϕ-Response Using Different Resolutions

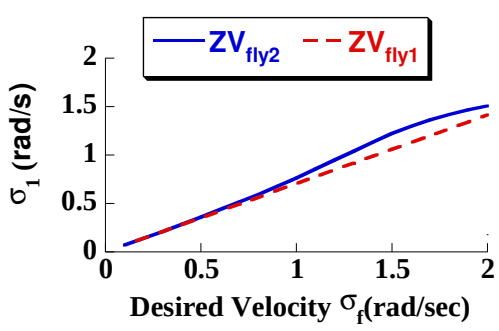


(b) Close-Up of ϕ-Response Using Different Resolutions

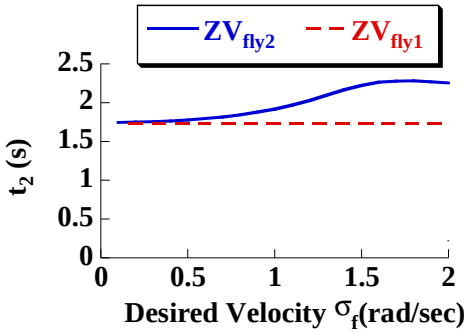
Figure 6.15: Numerical Errors in ZV_{fly2} Solution.

resolution the residual vibration is significantly lower, indicating that the residual vibration stems from numerical error.

The "shape" of the ZV_{fly1} and ZV_{fly2} commands can also be compared. Each command can be entirely described by the amplitude and duration, σ_1 and t_2, of the intermediate step, as was shown at the bottom of Figure 6.12. Figure 6.16(a) compares the amplitudes of the two shapers for various σ_f. The ZV_{fly1} solution gives a good approximation for the exact ZV_{fly2} solution for small $\sigma_f < 0.5$. This is to be expected because the ZV_{fly1} solution was developed from a linearized model that is only valid for small σ_f. Similarly, Figure 6.16(b) shows that the values for t_2 are approximately the same for $\sigma_f < 0.5$.



(a) Comparison of ZV_{fly1} and ZV_{fly2} Intermediate Step Amplitude.



(b) Comparison of ZV_{fly1} and ZV_{fly2} Intermediate Step Duration .

Figure 6.16: Comparison of ZV_{fly1} and ZV_{fly2} Parameters for L=3 m and R=1 m.

6.4 A Improved ZV Shaper for a Partially Linearized Fly-ball Model

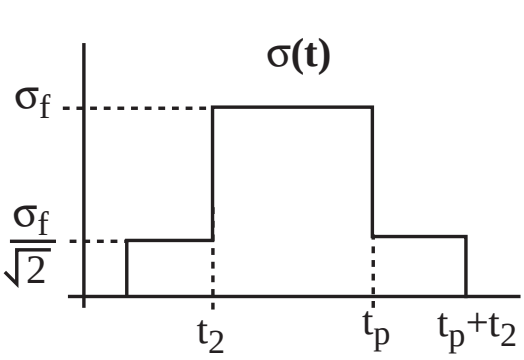
A refinement can be made to the ZV_{fly1} solution to make it a better approximation of the exact ZV_{fly2} solution. The purpose is to develop an analytical solution that performs better than the ZV_{fly1} solution. This will also give more insight into the relationship between the ZV_{fly2} parameters and σ_d shown in Figures 6.16(a) and 6.16(b).

Two constraints were used to linearize the equations of motion and derive the ZV_{fly1} shaper. Suppose the second constraint is eliminated; only a small angle approximation for ϕ is used. The equations of motion can be written as:

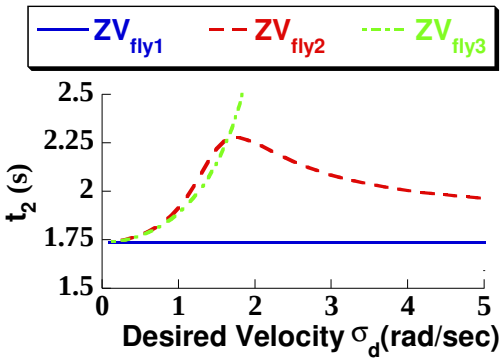
$$\ddot{\phi} + \left(\frac{g}{L} - \sigma^2\right)\phi = \frac{R}{L} \cdot \sigma^2 \tag{6.12}$$

In this form the natural frequency of the system is given by: $\sqrt{\frac{g}{L} - \sigma^2}$. Notice that the natural frequency varies as a function of the angular velocity σ. Because the natural frequency varies with σ, the shaped command should be designed to account for this variation.

Figure 6.17(a) gives a template for a ZV-shaped command. The amplitude of the intermediate step has been set to $\frac{\sigma_f}{\sqrt{2}}$, similar to a ZV_{fly1} command. However, the time duration of the intermediate step, t_2, is left unknown. Consider applying this command to the partially linearized system (6.12). Because σ is constant during the intermediate step of the



(a) ZV_{fly3} Command.



(b) Comparison of t_2 times for ZV_{fly1}, ZV_{fly2}, and ZV_{fly3} Shapers for L=3 m and R=1 m.

Figure 6.17: ZV_{fly3} Command and Comparison to Other Commands.

command, the natural frequency will be given by:

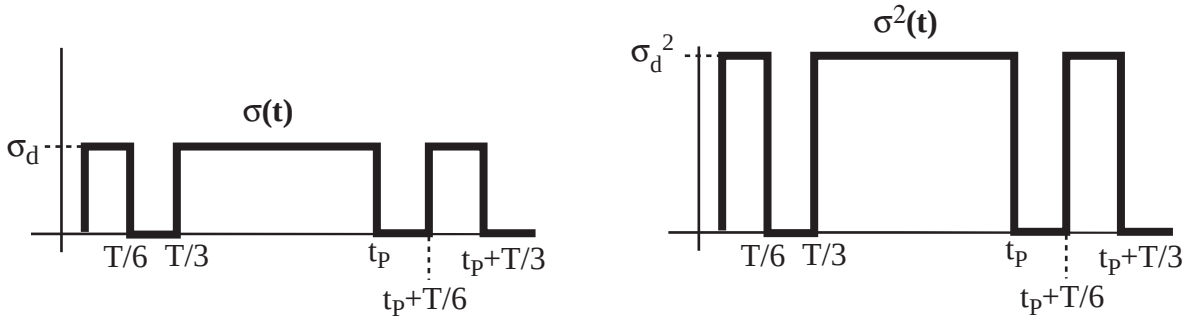
$$\begin{aligned} \omega_n &= \sqrt{\frac{g}{L} - \sigma^2} \\ &= \sqrt{\frac{g}{L} - \frac{\sigma_f^2}{2}} \end{aligned} \tag{6.13}$$

Recall that the time duration of the intermediate step of a shaped command should be equal to one-half the period. Therefore, t_2 should be chosen to be one-half the period of the frequency given by (6.13). A limitation to this technique is that it only applies to sufficiently long commands: $t_p > t_2$.

Using this method a new ZV shaped command is formed, called ZV_{fly3}. The command has the same amplitude profile as a ZV_{fly1} command, but a different time duration for the intermediate step. Figure 6.17(b) compares the t_2 times for the ZV_{fly1}, ZV_{fly2}, ZV_{fly3} solutions. Notice that the ZV_{fly3} shaper gives a much better numerical approximation of the ZV_{fly2} numerical solution for the low range of σ_d. This is because it accounts for the frequency correction discussed earlier.

6.5 A UMZV Shaper for a Partially Linearized Flyball model

Thus far, all of the shaped commands have been modeled on a ZV shaper. However, a different set of commands can be formed from a UMZV shaper template. UMZV commands have the advantage of being faster than a ZV shaped command and can also be implemented



(a) UMZV_{fly} Long Command

(b) UMZV_{fly} Long Command Squared.

Figure 6.18: Formation of UMZV_{fly} Command/

on on/off type drive systems. The UMZV commands developed for the flyball system will be called UMZV_{fly} commands. All such commands will be based off the linearized model (6.7), similar to the ZV_{fly1} shaper. As was the case earlier, the solution depends on the pulse time, t_p. In this case their are three different solution cases: $t_p < T/6$ is termed a *short command*, $T/6 < t_p < T/3$ is termed a *medium command*, and $T/3 < t_p$ is termed a *long command*. T is the linearized period given by $\frac{2\pi}{\sqrt{g/L}}$. Three ranges are needed because the UMZV shaped command has a different shape for each of the aforementioned conditions. The following subsections derive each of the three solutions.

6.5.1 UMZV_{fly} Long Commands

The UMZV_{fly} long command is the simplest of the three solutions, because it is identical to a standard UMZV command. Figure 6.18(a) shows a UMZV_{fly} long command. Recall that in the linearized equations of motion (6.7) this command is squared. Figure 6.18(b) shows this squared command. The only difference between Figures 6.18(a) and 6.18(b) is that the amplitude of the command is scaled by a constant factor. Because the original command was a UMZV command, and scaling is a linear operation, the scaled command will still eliminate residual vibration. Therefore, this is the UMZV_{fly} solution for $T/3 < t_p$.

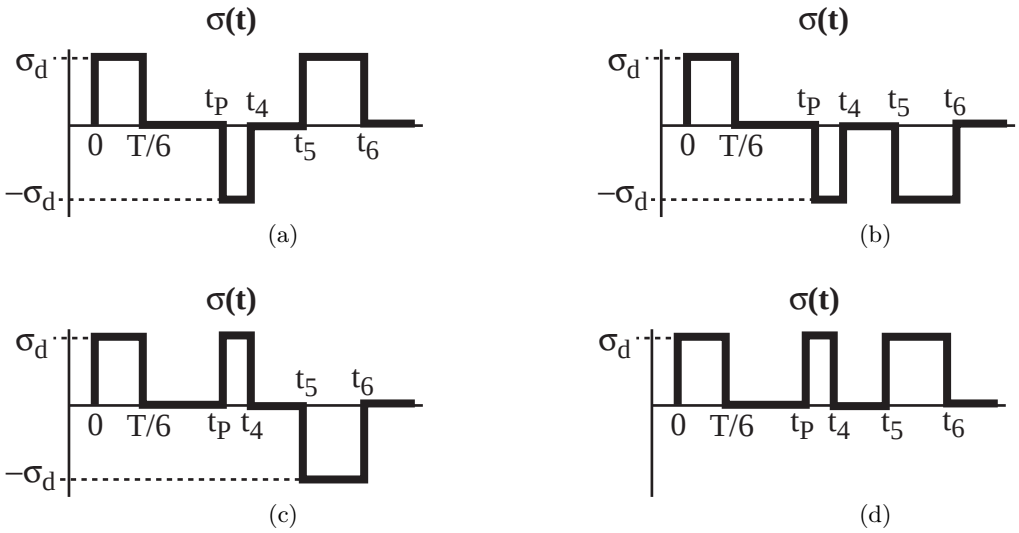


Figure 6.19: Possible Templates for the UMZV_{fly} medium command.

6.5.2 UMZV_{fly} Medium Commands

Creating UMZV_{fly} commands for $T/6 < t_p < T/3$ is not nearly as intuitive a process as for the previous cases. These commands must be created by numerically solving a nonlinear optimization problem. The first step is to establish the general form of the command. Figure 6.19 gives four possible forms for the command. Notice that the initial part of each command, from $0 < t < t_p$, is identical. This is a necessity because the value of t_p is not known ahead of time and the shaping algorithm does not know whether the command will be short, medium or long. This means that for $0 < t < T/6$ the short, medium, and long commands must be identical for the shaper to be used real-time. This is why the medium command between $0 < t < T/6$ is already pre-determined. Each of the four possibilities was tested as a possible solution. Of these four, only Figure 6.19(a) converged to a solution and was therefore used as the template for UMZV_{fly} medium commands.

As shown in Figure 6.19(a), the unknowns are the three switch times: t_4, t_5, t_6. Solving for the switch times was performed as a nonlinear optimization, using the Matlab function, *fmincon*. The constraints to this problem were:

1. **Zero-Vibration Constraint.** This can be written as:

$$\sum_{k=1}^{6} e^{2\pi i \frac{t_k}{T}} = 0 \qquad\qquad t_k = \begin{cases} 0, & \text{for k=1} \\ \frac{T}{6}, & \text{for k=2} \\ t_p, & \text{for k=3} \\ t_k, & \text{for k=4,5,6} \end{cases}$$

where T is the system oscillation period and t_k are the switch times from Figure 6.19(a). The real and imaginary parts of the above sum can be set equal to zero since the total sum is zero. This leads to the final form of the constraint:

$$\text{real}\left(\sum_{i=1}^{6} e^{2\pi i \frac{\tilde{t}_k}{T}}\right) = 0 \qquad\qquad \text{imag}\left(\sum_{i=1}^{6} e^{2\pi i \frac{\tilde{t}_k}{T}}\right) = 0 \tag{6.14}$$

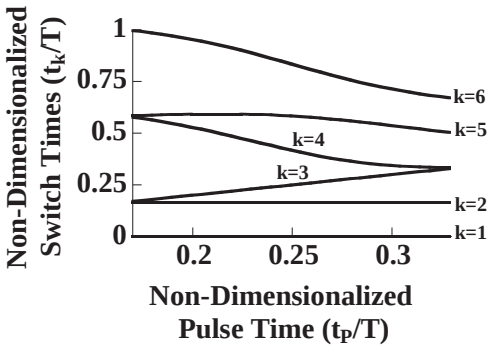
2. **Move Distance Proportional to t_p.** Both the long and short commands have a move distance equal to $\sigma_f \cdot t_p$. This same relationship should hold for medium commands. Setting this relationship equal to the area of the medium command in Figure 6.19(a) yields:

$$-\frac{T}{6} = -t_4 - t_5 + t_6 \tag{6.15}$$

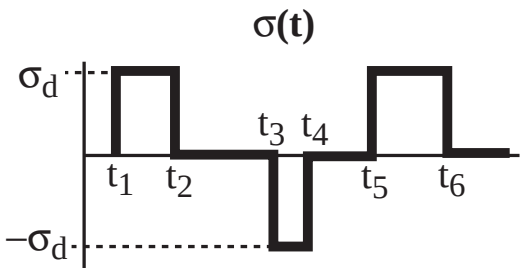
3. **Time Ordering.** The impulse times must be sequential:

$$t_4 \leq t_5 \leq t_6 \tag{6.16}$$

The value that was minimized during the optimization was the time of the final impulse, t_6. The values of t_4, t_5, t_6 were solved for $T/6 < t_p < T/3$. The first value of t_p used was: $t_p = T/3 - \Delta$, where delta was a small number (eg. $\Delta = 0.001$). Under these conditions it was assumed that the command would be similar to a UMZV_{fly} long command. Based on this assumption, an initial guess of $t_4 = t_p + \Delta, t_5 = t_p + \frac{T}{6}, t_6 = t_p + \frac{T}{3}$ was used. After a solution was found the problem was resolved for $t_p = T/3 - 2\Delta$, using the answer from the previous problem as the initial guess. This process was repeated, backward stepping from $t_p = T/3$ to $t_p = T/6$. The switch time solutions are shown in Figure 6.20(a). Each curve shows the value of t_k normalized by the period T. The horizontal axis shows the pulse time normalized by the period T. For reference, Figure 6.20(b) shows how the t_k times are used to formulate the UMZV_{fly} medium command.



(a) Switch Times for UMZV_{fly} Medium Command.



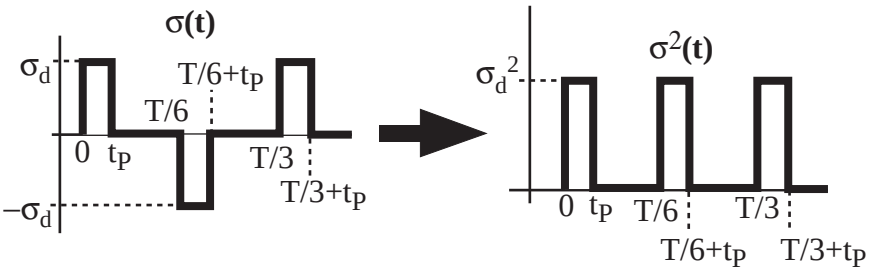
(b) UMZV_{fly} Medium Command Template.

Figure 6.20: UMZV_{fly} Medium Command for Various Pulse Times.

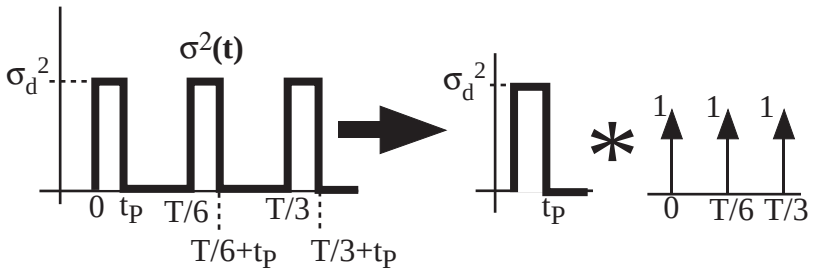
6.5.3 UMZV_{fly} Short Commands

The UMZV_{fly} short command is also similar to a standard UMZV short command, but with a few modifications. As a first guess, suppose a standard UMZV shaped pulse command is used (assuming $t_p < T/6$). The left hand side of Figure 6.21(a) shows such a command. The right hand side of the figure shows the command after it is squared. The squaring process has flipped the sign of the second pulse. The physical explanation for this is that the mass will swing outward (positive ϕ) regardless of whether σ is positive or negative. The question is: *Does this effect the ability of the command to eliminate residual vibration?* To answer this question a simple vector diagram analysis is used. The command is deconvolved into a step and an impulse sequence, as shown in Figure 6.21(b). The step command will certainly cause residual vibration. So the question is whether the impulse sequence generates zero residual vibration. The impulses can be mapped onto a vector diagram, as shown in Figure 6.21(c). Clearly these vectors do not add to zero. Therefore, the command will cause residual vibration.

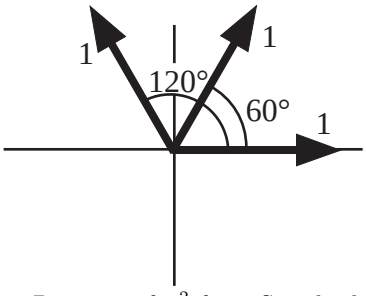
The vector diagram in Figure 6.21(c) showed that all three impulses were positive. This is because the original command was squared in Figure 6.21(a). In a standard UMZV vector diagram the impulses add to zero because the second vector is negative. Because the second vector cannot be made negative, it must be shifted by 180^o into the third-quadrant, as shown in Figure 6.21(d). The vectors in this new diagram clearly add to zero. This new



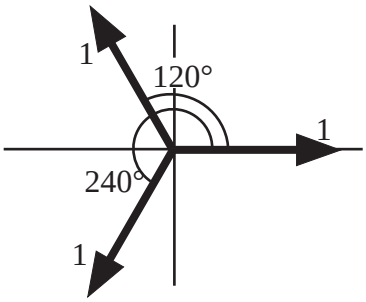
(a) Standard UMZV Short Command.



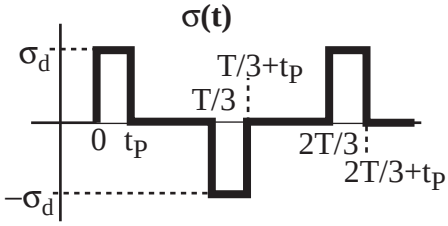
(b) Decomposition of σ^2 For a Standard UMZV Short Command.



(c) Vector Diagram of σ^2 for a Standard UMZV Short Command.



(d) Revised Vector Diagram for UMZV_{fly} Short Command.



(e) UMZV_{fly} Short Command.

Figure 6.21: Evolution of UMZV_{fly}Short Command.

vector diagram can be used to calculate the correct UMZV_{fly} short command, shown in Figure 6.21(e). However, this command will only work for $t_p < T/6$.

6.5.4 UMZV_{fly} Command Summary and Performance

Figure 6.22 shows the UMZV_{fly} switch times for short, medium, and long commands. There are six lines, each corresponding to a single switch time. The horizontal axis is non-dimensionalized pulse time. The vertical axis is the non-dimensionalized time of the impulses. The plot is divided into three regions corresponding to short, medium, and long commands. Above the plot is a small sketch of the template for each type of command. The line with circular markers shows the non-dimensionalized pulse time for reference. Notice that all switch times beneath this line stay constant, and are independent of the pulse time. This shows that the shaper can be used real-time and does not need to know the command length, t_p, before the command begins.

To illustrate how this might be implemented on a real system, consider the following example. As soon as the operator initiates a move the velocity switches on since the 1^{st} switch time is always at $t = 0$. If the operator commands the system to stop before $t = \frac{T}{6}$ then the system knows the command is *short* and can calculate the remaining switch times based on Figure 6.22. If this is not the case, then the system knows the command is at least as long as a *medium* command. Therefore at time $t = \frac{T}{6}$, the system switches to zero velocity, since the figure indicates this is the second switch time for all *medium* and *long* commands. If the operator commands the system stop before $t = \frac{T}{3}$ then the command is of *medium* length and the remaining switch times are determined by the figure. If this is not the case than the command must be *long*. As soon as $t = \frac{T}{3}$ then the velocity switches on since this is the third switch time for all *long* commands. The remaining switch times can then be determined by the figure. Using this type of *If-Then* type program structure, the command generator produces the UMZV_{fly} command without knowing the command length, t_p ahead of time.

Figure 6.23 shows the move distance as a function of the non-dimensionalized pulse time. The vertical axis shows the non-dimensionalized move distance, where θ is the angular

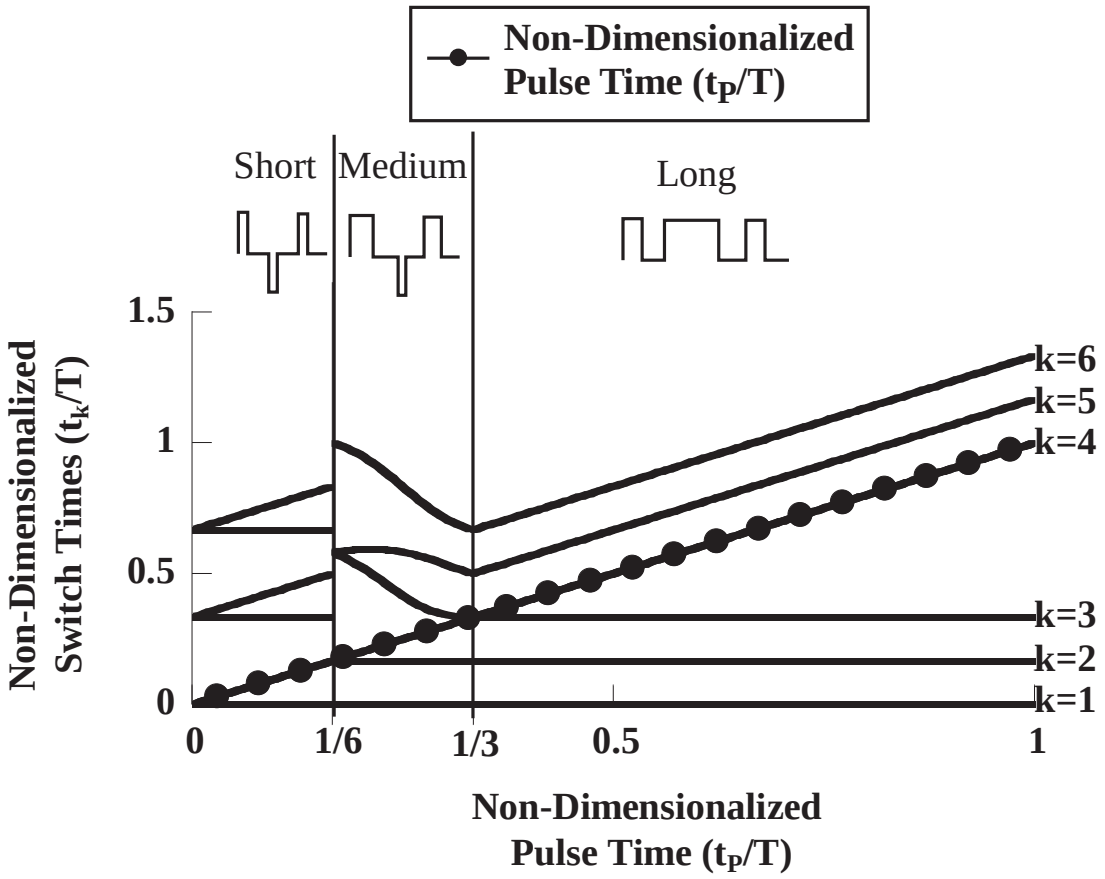


Figure 6.22: UMZV_{fly} Switch Times for Various t_p.

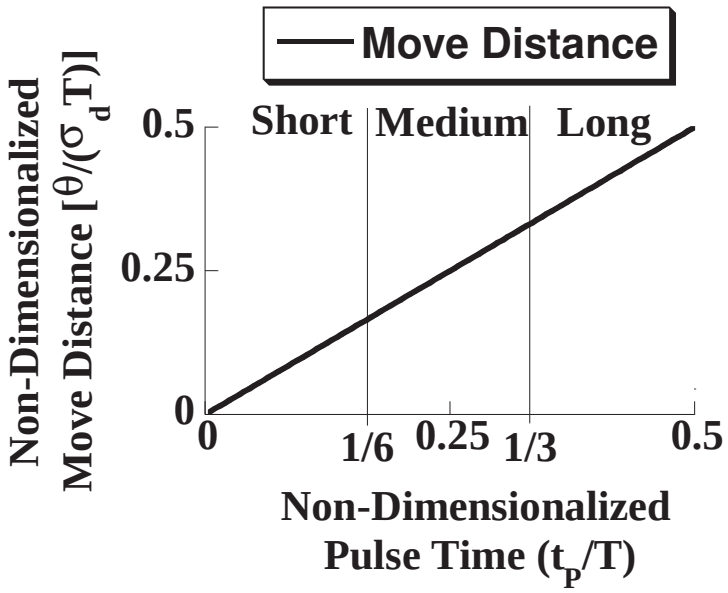
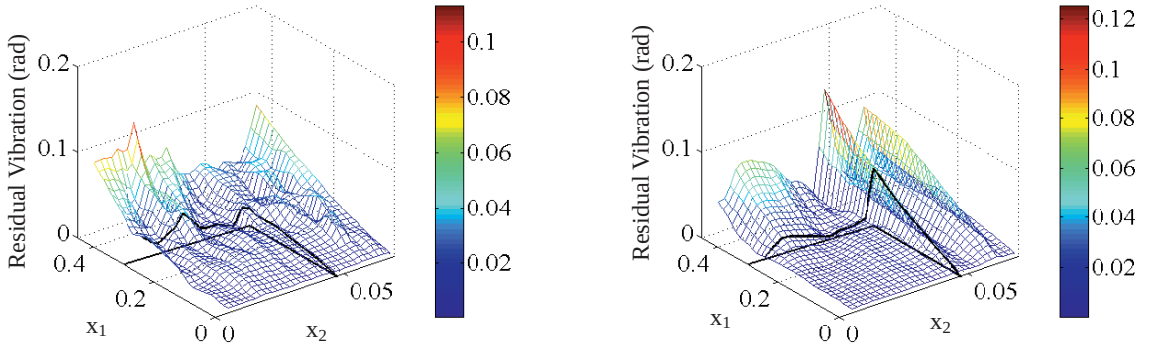


Figure 6.23: UMZV_{fly} Move Distance for Various t_p.



(a) UMZV_{fly} Residual Vibration.

(b) ZV_{fly1} Residual Vibration.

Figure 6.24: Comparison of Residual Vibration of UMZV_{fly} and ZV_{fly1} Commands for $\sigma_f = 0.25$ m/s.

displacement of the mast in radians. This graph shows that as the command transitions from short to medium to long, the move distance varies linearly with the pulse time. This is important from the operator's perspective because the move distance should be proportional to the pulse time.

The performance of the UMZV_{fly} shaper was evaluated by measuring its residual vibration across a range of parameters. Figure 6.24(a) shows the results of these simulations. Figure 6.24(b) is a plot of the ZV_{fly1} results shown earlier in Figure 6.11(c) and is duplicated here for easy comparison. The same parameter set and axes are used in both plots. The black line represents the boundary of the two assumptions for linearity. Inside the boundary the shaper performs relatively well. Notice that outside this boundary the UMZV_{fly} shaper performance degrades rapidly. Comparing the two figures, the UMZV_{fly} shaper induces slightly more vibration than the ZV_{fly1} shaper. The shaper performance also degrades faster as the parameters move from the linear region to the non-linear region. This is to be expected because UMZV shapers, in general, are more sensitive to modeling errors. However, the UMZV_{fly} residual vibration is far superior to the unshaped case shown earlier in Figure 6.11(a).

The duration UMZV_{fly} command can also be compared to the ZV_{fly1} command. Figure 6.25 shows the duration of both these commands for various pulse times, t_p. The vertical

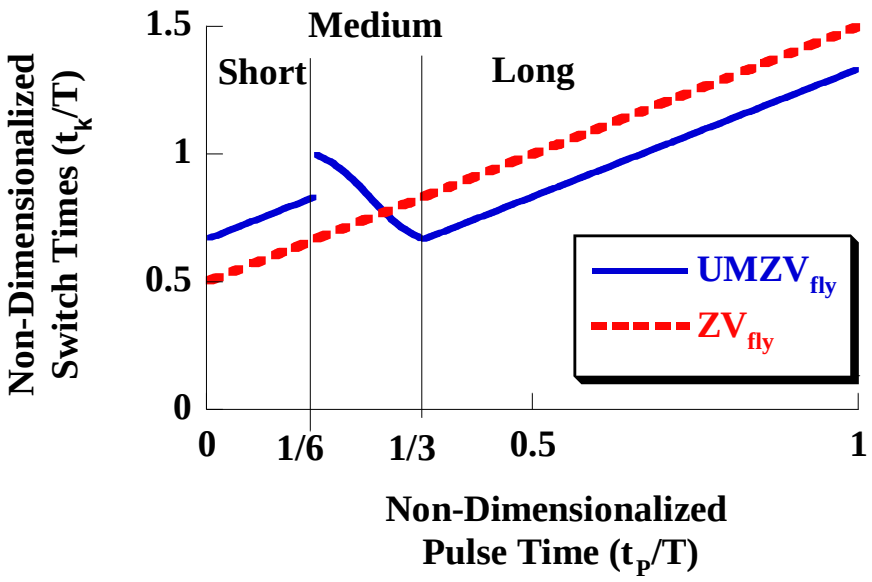


Figure 6.25: Comparison of Duration of UMZV$_{\text{fly}}$and ZV$_{\text{fly1}}$commands for various t_p.

axis is normalized by the period as in the previous plots. The command duration was measured using the final switch time of each shaper. Therefore, for the ZV$_{\text{fly1}}$ $k = 2$ since there are only two switch times, whereas for the UMZV$_{\text{fly}}$ $k = 6$ since there are six switch times. For long commands the UMZV$_{\text{fly}}$ command has a shorter command duration than the ZV$_{\text{fly1}}$. However, for some *medium* commands and all *short* commands the ZV$_{\text{fly1}}$ command is shorter. It should be noted that the "space" of *long* commands extends infinitely in the positive direction along the horizontal axis. So, for most of the operating space of the system a UMZV$_{\text{fly}}$command will have a duration than the ZV$_{\text{fly1}}$ command.

Considering the residual vibration and command characteristics of ZV$_{\text{fly1}}$ and UMZV$_{\text{fly}}$ commands, the UMZV$_{\text{fly}}$ command has several advantages. It has a low residual vibration that is comparable to the ZV$_{\text{fly1}}$ command and has a shorter duration for *long* commands. Considering that these solutions are aimed at tower cranes, most of the commands will involve moving the payload across a large distance and will therefore be categorized as *long*. Even if *short* or *medium* commands are used the added time is still relatively small. If it is known ahead of time that the majority of the motions will be *short* than the UMZV$_{\text{fly}}$ algorithm could be switched with the ZV$_{\text{fly1}}$ algorithm. The ZV$_{\text{fly1}}$ algorithm also has the

advantage of being simpler than the UMZV_{fly} algorithm.

6.6 Flyball Study Conclusions

This section has addressed the problem of creating shaped commands for a 1-mass flyball system. Initially, a model was developed using standard Newton-Euler analysis. Three different ZV shaped commands were derived: the ZV_{fly1} and ZV_{fly3} shapers were derived from the partially linearized equations of motion while, the ZV_{fly2} shaper was derived numerically. In addition, a unity-magnitude command was also developed using a partially linearized model. All of the shapers were tested in simulation across a wide parameter space. The limitations of each shaper within this parameter space were discussed.

It should be noted that the techniques developed in this chapter could be extended to develop many more shaped commands for the 1-mass flyball system. For example, both ZVD and EI shaped commands have a two-step staircase profile for sufficiently long pulse times. The amplitude of each step could be adjusted in anticipation of the squaring effect in the system dynamics, similar to the process used for the ZV_{fly1} command.

This chapter has also revealed several important concepts for designing shapers for rotating systems provided there is no radial motion:

- Linearization can reveal important information about the system's natural frequencies.
- While the system is spinning at constant velocity, there exists a steady-state deflection angle, ϕ_{ss}. When using a shaped command, the deflection is equal to this steady state angle when the angular velocity has reached its final setpoint value. However this is only true for sufficiently long commands.
- The previous observation can be combined with numerical "shooting" strategies to solve for shaper times and amplitudes.
- The move distance can change the shaper solution.
- Increasing either the velocity σ_f, arm length R, or linkage length L tends to make the system behave more non-linearly.

These observations will be used in the design of input shapers for tower cranes in the next chapter.

CHAPTER VII

INPUT SHAPING FOR TOWER CRANES

Of the various types of cranes, tower cranes are some of the most commonly used. Because of their simple mast-and-jib structure they can be easily broken-down, transported, and reconstructed. Another big advantage to these cranes is that they can have a very large workspace, but only occupy a relatively small footprint on the ground. Because of these properties tower cranes used most extensively in construction sites, but can also appear in warehouses and factories on a smaller scale.

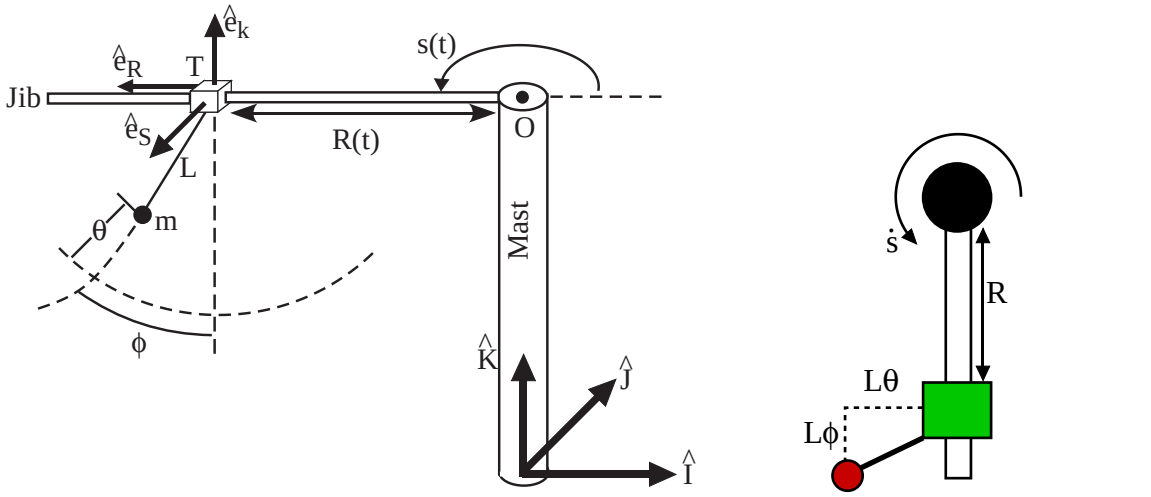
Despite their widespread use, input shaping has not been widely applied to tower cranes. The main reason for this is the nonlinear dynamics of tower cranes. The rotational, or slewing, motion causes centripetal and Coriolis acceleration that lead to nonlinear, coupled equations of motion. Standard input shaping is not designed for these types of systems.

This chapter will focus on developing input shapers for tower cranes. First, the equations of motion for a tower crane will be derived and analyzed. Second, two new shaping strategies for tower cranes will be presented. Third, these new strategies will be tested against standard shaping and unshaped commands in simulations and experiments. Many of the techniques in this chapter can be extended to other rotational-type cranes (eg. boom cranes) as well as rotating machinery in general.

7.1 Tower Crane Equations of Motion

Figure 7.1(a) shows a detailed sketch of a tower crane. The trolley position, point T, is described in cylindrical coordinates; $R(t)$ is the radial position, and $s(t)$ is the slew angle. The inertial coordinate system $\hat{I}, \hat{J}, \hat{K}$ is fixed to ground. A second coordinate system, $\hat{e}_R, \hat{e}_s, \hat{e}_k$ is fixed to the trolley. Note that $\hat{e}_R$ always points in the direction of increasing radial length $R(t)$, and $\hat{e}_s$ points in the direction of increasing slew angle $s(t)$.

A mass, m, hangs from the trolley at a fixed length of L. The payload position is



(a) Tower Crane Side View.

(b) Tower Crane Overhead View.

Figure 7.1: Tower Crane Sketch: Side View and Overhead View.

measured by two spherical coordinates: ϕ and θ. Note that the payload is assumed to be a point mass so the payload twisting about the suspension cable is ignored. These two coordinates describe the mass position using two body-fixed rotations about the $\hat{e}_R, \hat{e}_s, \hat{e}_k$ coordinate system. The non-rotated position is the mass hanging vertically down at a distance L. The first rotation is the θ rotation about $-\hat{e}_R$. If the same rotation were applied to the $\hat{e}_R, \hat{e}_s, \hat{e}_k$-frame a new rotated coordinate system $\hat{e}'_R, \hat{e}'_s, \hat{e}'_k$ is formed(not shown in figure). The second rotation is the ϕ rotation about $-\hat{e}'_s$ from second coordinate system.

To get a better sense of the ϕ and θ coordinates, Figure 7.1(b) shows an overhead view of the tower crane. The green rectangle is the trolley and the red circle is the payload. For small angles the ϕ and θ coordinates are related to the physical payload coordinates in the following manner: $L\theta$ represents the payload displacement from the trolley perpendicular to the jib, $L\phi$ represents the payload displacement from the trolley parallel to the jib.

The ϕ and θ coordinates are closely related to standard spherical coordinates, but they are *not* identical. The relationship between these two coordinate systems will become useful for deriving the acceleration of mass m. The standard definition of spherical coordinates is given in [22]. Suppose $\hat{\imath}_{sp}, \hat{\jmath}_{sp}, \hat{k}_{sp}$ represents the standard cartesian coordinate system used

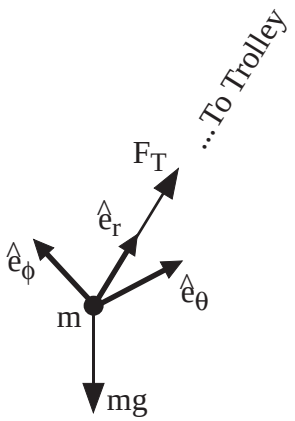


Figure 7.2: Free-Body Diagram of Payload

for defining spherical coordinates. These coordinates can be aligned with the trolley coordinate system: $\hat{e}_R = -\hat{k}_{sp}, \hat{e}_s = -\hat{\jmath}_{sp}, \hat{e}_k = -\hat{\imath}_{sp}$. Then the relationship between standard spherical coordinates (ϕ_{sp} and θ_{sp}) and the coordinates used to describe mass m (ϕ and θ) is as follows: $\phi_{sp} = \phi + \frac{\pi}{2}, \theta_{sp} = \theta$.

Figure 7.2 shows another coordinate system that will be useful for the derivation of the equations of motion. The $\hat{e}_\theta, \hat{e}_\phi, \hat{e}_r$ coordinate system is fixed to the mass: $\hat{e}_\theta$ points in the direction of increasing θ, $\hat{e}_\phi$ points in the direction of increasing ϕ, and $\hat{e}_r$ is aligned with the cable and points toward the trolley. The transformation between the mass coordinate system, $\hat{e}_\theta, \hat{e}_\phi, \hat{e}_r$, and the trolley coordinate system,$\hat{e}_R, \hat{e}_s, \hat{e}_k$, is given by:

$$\begin{bmatrix} \hat{e}_R \\ \hat{e}_S \\ \hat{e}_K \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ \sin\phi\sin\theta & -\cos\theta & \sin\theta\cos\phi \\ \sin\phi\cos\theta & \sin\theta & \cos\phi\cos\theta \end{bmatrix} \cdot \begin{bmatrix} \hat{e}_\phi \\ \hat{e}_\theta \\ \hat{e}_r \end{bmatrix} \tag{7.1}$$

When analyzing results, it will be useful to plot the position of the payload and trolley relative to the origin, O, shown at the top of the mast in 7.1(a). The cartesian coordinates of the trolley are given by:

$$x_T = R\cos s \tag{7.2}$$

$$y_T = R\sin s \tag{7.3}$$

where x_T and y_T are the cartesian coordinates of the trolley with respect to $\hat{I}, \hat{J}, \hat{K}$. The

cartesian coordinates of the payload are given by:

$$x_m = R\cos s + L\sin\phi\cos s + L\sin\theta\cos\phi\sin s \tag{7.4}$$

$$y_m = R\sin s + L\sin phi\sin s - L\sin\theta\cos\phi\cos s \tag{7.5}$$

It will also be useful to plot the payload response in polar coordinates relative to origin O:

$$R_m = \sqrt{x_m^2 + y_m^2} \tag{7.6}$$

$$\Theta_m = \tan^{-1}\frac{y_m}{x_m} \tag{7.7}$$

where R_m and Θ_m are the radial and rotational polar coordinates of the mass, and x_m and y_m are the cartesian coordinates of the mass given in (7.4) and (7.5). Θ_m is measured as the angle between the jib and $\hat{I}$ in the counter-clockwise direction (positive $\hat{K}$ rotation). The polar coordinates of the trolley are simply R and s.

Although the coordinate systems used in this problem may seem complicated, each one was carefully chosen to make the solution simpler. The trolley is described using cylindrical coordinates, and the mass is described using spherical coordinates. Since these are both standard coordinate systems the position, velocity, and acceleration of a particle are already known, and given in [22]. This makes the derivation of the equations of motion simpler. Also, note that using standard spherical coordinates produces a singularity at $\phi_{sp} = 0, \pi$ since changing θ_{sp} does not change the position of the particle being described. Therefore, the spherical coordinates ϕ and θ were chosen such that this singularity is outside the solution space. In some published works [3, 67] this precaution was not taken, and as a result there is a singularity at equilibrium in the equations of motion.

7.1.1 Derivation of Equations of Motion

Using the parameters and coordinates defined in the previous section, the equations of motion for the tower crane can be derived. Figure 7.2 shows a freebody diagram of the mass. Applying the Newton-Euler equations to this system yields:

$$\sum \vec{F} = m\vec{a}_m \tag{7.8}$$

$$\vec{F}_T \cdot \hat{e}_r - mg \cdot \hat{e}_k = m\vec{a}_m$$

where $\vec{a}_m$ is the acceleration of m, and $\vec{F}_T$ is the tension from the cable.

The acceleration of m can be derived by utilizing the moving reference frame $\hat{e}_R, \hat{e}_s, \hat{e}_k$ attached to the trolley T. This process is similar to the flyball derivation given in the previous chapter. The equation that relates the acceleration of m to the moving reference frame at T is [22]:

$$a_m = \vec{a}_T + \vec{a}_{m/T} + 2\vec{\omega}_T \times \vec{v}_{m/T} + \vec{\alpha}_T \times \vec{r}_{m/T} + \vec{\omega}_T \times \vec{\omega}_T \times \vec{r}_{m/T} \tag{7.9}$$

where,

$$\begin{aligned}
\vec{a}_T &= \text{Acceleration of T} \\
\vec{\omega}_T &= \text{Angular Velocity of T-frame}(\hat{e}_R, \hat{e}_s, \hat{e}_k) \\
\vec{\alpha}_T &= \text{Angular Acceleration of T-frame}(\hat{e}_R, \hat{e}_s, \hat{e}_k) \\
\vec{r}_{m/T} &= \text{Position vector from T to m} \\
\vec{v}_{m/T} &= \text{Velocity vector of m relative to T} \\
\vec{a}_{m/T} &= \text{Acceleration of m relative to T}
\end{aligned}$$

These terms can be defined using the parameters shown in Figure 7.1(a):

$$\begin{aligned}
\vec{a}_T &= (\ddot{R} - R\dot{s}^2) \cdot \hat{e}_R + (R\ddot{s} + 2\dot{R}\dot{s}) \cdot \hat{e}_s \\
\vec{\omega}_T &= \dot{s} \cdot \hat{e}_k \\
\vec{\alpha}_T &= \ddot{s} \cdot \hat{e}_k \\
\vec{r}_{m/T} &= -L \cdot \hat{e}_r \\
\vec{v}_{m/T} &= L\left(\dot{\phi} \cdot \hat{e}_\phi + \dot{\theta}\cos\phi \cdot \hat{e}_\theta\right) \\
\vec{a}_{m/T} &= L\left((\ddot{\phi} + \dot{\theta}^2 \cos\phi \sin\phi) \cdot \hat{e}_\phi + (\ddot{\theta}\cos\phi - 2\dot{\phi}\dot{\theta}\sin\theta) \cdot \hat{e}_\theta\right)
\end{aligned} \tag{7.10}$$

Note that the terms $\vec{r}_{m/T}$, $\vec{v}_{m/T}$, and $\vec{a}_{m/T}$ can be derived from the position, velocity, and acceleration of a particle in spherical coordinates. The term $\vec{a}_T$ can be derived from the acceleration of a particle in cylindrical coordinates. All of these terms can be found in [22]. It is also assumed that the suspension length, L, is constant.

To solve for the equations of motion, all of the terms in (7.10) are transformed to a common coordinate system. In this case (7.1) can be used to transform everything into the mass-fixed coordinate system $\hat{e}_\theta, \hat{e}_\phi, \hat{e}_r$. Then, these terms can be substituted back into (7.9) to solve for the total acceleration of the mass. The total acceleration can then be substituted

into (7.8). The mass cannot move in the direction of the cable, $\hat{e}_r$. Therefore, all terms involving $\hat{e}_r$ can be dropped from the force balance. The remaining vector equation can be separated into two equations by evaluating the components of each vector. Evaluating the components of $\hat{e}_\theta$ yields one equation of motion:

$$\begin{aligned} &L\ddot{\theta}\cos\phi - 2L\dot{\phi}\dot{\theta}\sin\phi + g\sin\theta = \\ &\quad R\ddot{s}\cos\theta + 2\dot{R}\dot{s}\cos\phi + 2L\dot{s}\dot{\phi}\cos\phi\cos\theta + L\ddot{s}\sin\phi\cos\theta + \\ &\quad L\dot{s}^2\sin\theta\cos\phi\cos\theta \end{aligned} \tag{7.11}$$

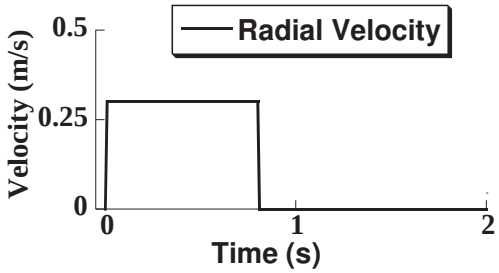
A second equation of motion is formed by evaluating the components of $\hat{e}_\phi$:

$$\begin{aligned} &L\ddot{\phi} + L\dot{\theta}^2\cos\phi\sin\phi + g\sin\phi\cos\theta = \\ &\quad -\ddot{R}\cos\phi + R\dot{s}^2\cos\phi - R\ddot{s}\sin\phi\sin\theta - 2\dot{R}\dot{s}\sin\phi\sin\theta - \\ &\quad 2L\dot{s}\dot{\theta}\cos^2\phi\cos\theta - L\ddot{s}\sin\theta + L\dot{s}^2\sin\phi\cos^2\theta\cos\phi \end{aligned} \tag{7.12}$$

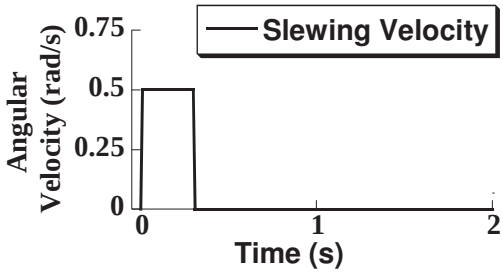
7.1.2 Comparison of Equations of Motion with Literature

To test the validity of the derived equations of motion, they were compared to an existing model in the literature. Golafshani presents a model for a tower crane in [23]. However, this model is based on a totally different set of coordinates for measuring the position of the mass. Therefore a direct, term-by-term comparison was not possible. Instead, computer simulations were run using both models and the absolute position of the payload was compared.

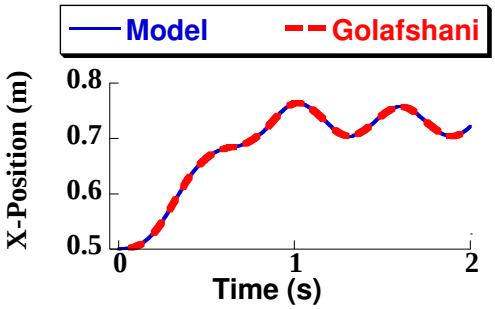
Figure 7.3 compares the system response using the derived equations of motion (7.11) and (7.12) with the system response using Golafshani's model. Figures 7.3(a) and 7.3(b) show the commanded radial and slewing velocity respectively. Both commands were trapezoidal velocity profiles. Figures 7.3(c) and 7.3(d) give the payload response. The solid line is the response from the derived model, while the dashed line is Golafshani's response. The figures clearly show that the two models give an identical response. Several trials were simulated using different commands. In all cases the responses matched perfectly. This helps validates the derived equations of motion (7.11) and (7.12). Experimental results presented in later sections also validate the model.



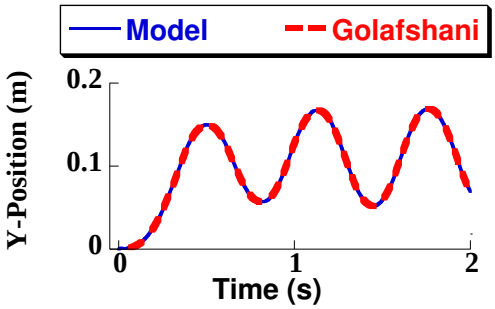
(a) Radial Velocity.



(b) Slewing Velocity.



(c) Payload X-Response.



(d) Payload Y-Response.

Figure 7.3: Equations of Motion Test: Derived Model vs. Golafshani's Model

7.2 Analysis and Linearization of Tower Crane Equations

The equations of motion developed in the previous section fully describe the dynamics of the system. This section strives to "unlock" more information from the equations of motion to get a better understanding for the system dynamics.

7.2.1 Radial Motion

Consider a purely radial motion. Under these conditions $s = \dot{s} = \ddot{s} = 0$. Substituting these conditions into (7.11) and (7.12) yields:

$$L\ddot{\theta}\cos\phi - 2L\dot{\phi}\dot{\theta}\sin\phi + g\sin\theta = 0 \tag{7.13}$$

$$L\ddot{\phi} + L\dot{\theta}^2\cos\phi\sin\phi + g\sin\phi\cos\theta = -\ddot{R}\cos\phi \tag{7.14}$$

Because the right-hand-side of (7.13) is zero, it follows that $\theta = \dot{\theta} = \ddot{\theta} = 0$. Substituting this into (7.14) yields:

$$L\ddot{\phi} + g\sin\phi = -\ddot{R}\cos\phi \tag{7.15}$$

which is the equation of motion for a planar pendulum with a moving base. This makes sense intuitively because for a pure radial motion and no rotation, the tower crane is a planar pendulum.

Input shaping for a planar pendulum has already been studied extensively. Therefore, this problem will not be discussed further in this thesis and the rest of this section will focus on pure rotational motion.

7.2.2 Rotational Motion

Pure rotational motion implies that $\dot{R} = \ddot{R} = 0$. Substituting this into the equations of motion yields:

$$L\ddot{\theta}\cos\phi - 2L\dot{\phi}\dot{\theta}\sin\phi + g\sin\theta = \tag{7.16}$$
$$R\ddot{s}\cos\theta + 2L\dot{s}\dot{\phi}\cos\phi\cos\theta + L\ddot{s}\sin\phi\cos\theta + L\dot{s}^2\sin\theta\cos\phi\cos\theta$$

$$L\ddot{\phi} + L\dot{\theta}^2\cos\phi\sin\phi + g\sin\phi\cos\theta = \tag{7.17}$$
$$R\dot{s}^2\cos\phi - R\ddot{s}\sin\phi\sin\theta - 2L\dot{s}\dot{\theta}\cos^2\phi\cos\theta - L\ddot{s}\sin\theta + L\dot{s}^2\sin\phi\cos^2\theta\cos\phi$$

Analyzing these equations of motion is the focus of the following sections.

7.2.3 Steady State Deflection

In the flyball analysis it was shown that given a steady-state angular velocity σ_{ss}, there existed a steady-state deflection ϕ_{ss}. The same approach will be taken with the tower crane: *Given a steady-state angular velocity, $\dot{s}_{ss}$, find the steady-state deflection of the payload.* As was the case with the flyball study, this will be useful for understanding system response and deriving appropriately shaped commands. In addition, ϕ_{ss} will be used later for linearizing the equations of motion.

To find the steady-state deflection of the system, it is assumed that the tower crane is rotating at constant angular velocity: $\dot{s} = \dot{s}_{ss}, \ddot{s} = 0$. The conditions for steady state are

zero velocity and acceleration of the deflection angles: $\dot{\phi} = \dot{\theta} = \ddot{\phi} = \ddot{\theta} = 0$. Substituting these assumptions into the equations of motion (7.16) and (7.17) yields:

$$g \sin\theta_{ss} = L\dot{s}_0^2 \sin\theta_{ss} \cos\phi_{ss} \cos\theta_{ss} \tag{7.18}$$

$$g \sin\phi_{ss} \cos\theta_{ss} = R\dot{s}_{ss}^2 \cos\phi_{ss} + L\dot{s}_{ss}^2 \sin\phi_{ss} \cos^2\theta_{ss} \cos\phi_{ss} \tag{7.19}$$

where θ_{ss} and ϕ_{ss} are the steady-state deflection angles.

To satisfy (7.18) we can set:

$$\theta_{ss} = 0 \tag{7.20}$$

If this value is substituted into (7.18), both sides become zero thus satisfying the equality. Substituting this into the second equation (7.19) yields:

$$g \sin\phi_{ss} = R\dot{s}_{ss}^2 \cos\phi_{ss} + L\dot{s}_{ss}^2 \sin\phi_{ss} \cos\phi_{ss}$$

or

$$0 = x_1 \cos(\phi_{ss}) + x_2 \sin(\phi_{ss}) \cos(\phi_{ss}) - \sin(\phi_{ss}) \tag{7.21}$$

$$x_1 = \left(\frac{R\dot{s}_{ss}^2}{g}\right) \qquad\qquad x_2 = \left(\frac{L\dot{s}_{ss}^2}{g}\right)$$

Notice that the above equation for ϕ_{ss} is identical to (6.5), the steady-state equation for the flyball system. Solutions to this equation were given in 6.4. In addition, as previously shown for small angles, $\phi_{ss} < 0.3$, the solution can be approximated as:

$$\begin{aligned} \phi_{ss} &= \frac{x_1}{1 - x_2} \\ &= \frac{R\dot{s}_{ss}^2}{g - L\dot{s}_{ss}^2} \end{aligned} \tag{7.22}$$

where x_1 and x_2 are the non-dimensionalized parameters previously shown in (7.21).

To summarize, the steady-state deflection of the system is given by $\theta_{ss} = 0$ and $\phi_{ss} = \frac{x_1}{1-x_2}$ (for small angles). Physically, this corresponds to the payload deflected radially outward in the plane formed by the mast and jib. According to the linearized approximation given in (7.22), as the slewing speed $\dot{s}$, trolley radial position R, or suspension length L increase, the steady-state deflection ϕ_{ss} will also increase. However, θ_{ss} is unaffected by any of these parameters.

7.2.4 Linearization

More insight can be gained by linearizing the equations of motion (7.16) and (7.17). The equations are linearized by considering small deviations from steady state, thus yielding:

$$\begin{aligned} \ddot{\phi} &= \left(\dot{s}_{ss}^2 - \frac{g}{L}\right)\phi - 2\dot{s}_{ss}\dot{\theta} \\ \ddot{\theta} &= \left(\dot{s}_{ss}^2 - \frac{g}{L}\right)\theta + 2\dot{s}_{ss}\dot{\bar{\phi}} \end{aligned} \tag{7.23}$$

where,

$$\bar{\phi} = \phi - \phi_{ss} \tag{7.24}$$

The new coordinate, $\bar{\phi}$ represents the deviation of ϕ from equilibrium, ϕ_{ss}, which is the solution to (7.21).

The linearized equations (7.23) can be put into state-space form:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\bar{\phi}} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \left(\dot{s}_{ss}^2 - \omega_0^2\right) & 0 & 0 & 2\dot{s}_{ss} \\ 0 & 0 & 0 & 1 \\ 0 & -2\dot{s}_{ss} & \left(\dot{s}_{ss}^2 - \omega_0^2\right) & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \bar{\phi} \\ \dot{\phi} \end{bmatrix} \tag{7.25}$$

where $\omega_0 = \sqrt{g/L}$ is the natural frequency of a pendulum with length L.

The eigenvalues of the linearized system are given by:

$$\begin{aligned} \lambda_{1,2} &= \pm\left(\omega_0 + \dot{s}_{ss}\right) i \\ \lambda_{3,4} &= \pm\left(\omega_0 - \dot{s}_{ss}\right) i \end{aligned} \tag{7.26}$$

The eigenvalues reveal that the linearized system is undamped and has two natural frequencies given by:

$$\omega_{1,2} = \omega_0 \pm \dot{s}_{ss} \tag{7.27}$$

Figure 7.4 shows a plot of the linearized natural frequencies as a function of slew velocity $\dot{s}_{ss}$. The dotted line shows ω_0, which is assumed to be one for this example. At zero velocity the natural frequency of the system is equivalent to the natural frequency of a pendulum of length L, $\omega_0 = \sqrt{g/L}$. As the slewing velocity increases the frequencies bifurcate, but remain centered about ω_0.

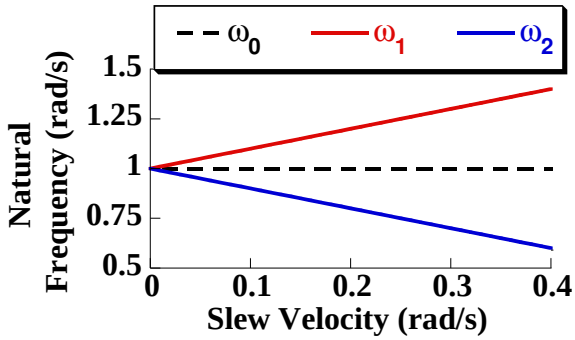


Figure 7.4: Linearized Frequencies for Various Slew Velocities

The state space equations (7.25) show that the ϕ and θ coordinates are coupled. The eigenvalue analysis revealed the presence of two modes. The two modes can be decoupled by finding a coordinate transformation, T, that transforms the state space equations into Jordan form. This transformation matrix is given by:

$$T = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -(\omega_0 + \dot{s}_{ss}) & 0 & (\omega_0 - \dot{s}_{ss}) \\ 0 & -1 & 0 & -1 \\ (\omega_0 + \dot{s}_{ss}) & 0 & (\omega_0 - \dot{s}_{ss}) & 0 \end{bmatrix} \tag{7.28}$$

and its inverse is:

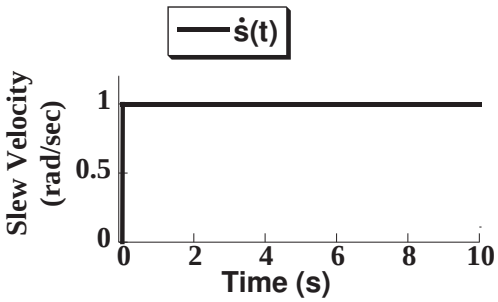
$$T^{-1} = \frac{1}{2} \begin{bmatrix} -1 + \frac{\dot{s}_{ss}}{\omega_0} & 0 & 0 & \frac{1}{\omega_0} \\ 0 & \frac{-1}{\omega_0} & -1 + \frac{\dot{s}_{ss}}{\omega_0} & 0 \\ 1 + \frac{\dot{s}_{ss}}{\omega_0} & 0 & 0 & \frac{1}{\omega_0} \\ 0 & \frac{1}{\omega_0} & -\left(1 + \frac{\dot{s}_{ss}}{\omega_0}\right) & 0 \end{bmatrix} \tag{7.29}$$

This transformation can be used to find new state variables, $z_1, \ldots, z_4$, from the operation:

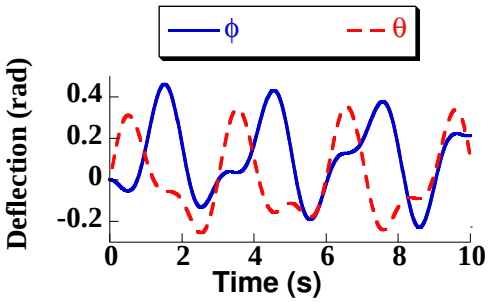
$$[\theta \ \dot{\theta} \ \bar{\phi} \ \dot{\phi}]^T = T \cdot [z_1 \ z_2 \ z_3 \ z_4]^T \tag{7.30}$$

Using these new state variables the equations of motion can be put into Jordan form:

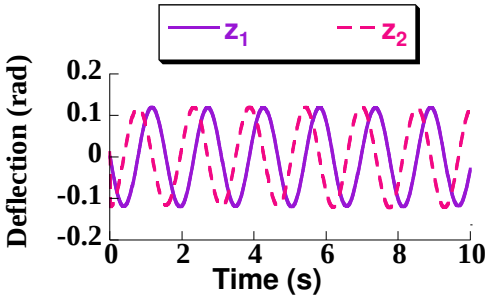
$$\dot{\vec{z}} = \begin{bmatrix} 0 & \omega_1 & 0 & 0 \\ -\omega_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega_2 \\ 0 & 0 & -\omega_2 & 0 \end{bmatrix} \cdot \vec{z} \tag{7.31}$$



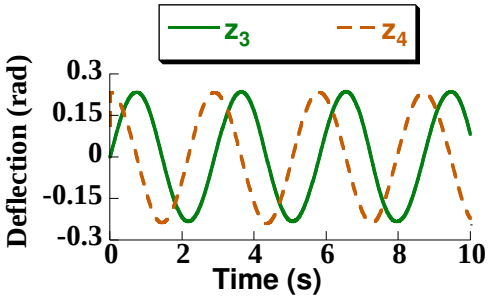
(a) Step Command.



(b) ϕ and θ Response.



(c) z_1 and z_2 Response.



(d) z_3 and z_4 Response.

Figure 7.5: Constant Slew Velocity Simulation Results [R=1, L=1].

where $\vec{z} = [z_1 \ z_2 \ z_3 \ z_4]^T$.

To test these theoretical results, a step response was simulated, as shown in Figure 7.5. Figure 7.5(a) shows the slewing command, a step in angular velocity. During the constant velocity portion of the command the system should be well approximated by the linearized equations (7.25). Figure 7.5(b) shows the θ and ϕ response. The responses clearly indicate a multi-mode response. To isolate the two modes, the ϕ-θ coordinates can be transformed to the $\vec{z}$ coordinates using the transformation given in (7.28) and (7.30). The response of the $\vec{z}$-coordinates is given in Figures 7.5(c) and 7.5(d). Notice that each plot shows a single-mode response, indicating the transformation successfully isolates the modes. $z_{1,2}$ represent the high frequency mode, and $z_{3,4}$ represent the low frequency mode. According to the figure the high frequency mode is given by $\omega_{high} = 4.13(rad/s)$ and the low frequency mode is $\omega_{low} = 2.12(rad/s)$. The pendulum frequency is $\omega_n = \sqrt{g}/L = 3.13$. Notice that this pendulum frequency is halfway between the high and low modes and the difference is given by $\dot{s}_f = 1$, exactly as predicted by (7.27). Finally, notice that the amplitude of z_1

and z_2 is exactly the same, and the two responses are shifted by 90°. The same holds for z_3 and z_4. This is typical for the undamped response of a system in Jordan form and also confirms the validity of the transformation.

7.3 Input Shaping for Tower Cranes

The last section showed that slewing motion, as opposed to radial motion, causes a nonlinear dynamic response in tower cranes. Therefore, this section will focus on: *How to rotate the crane without inducing residual vibration.* This section will discuss several different shaping strategies to solve this problem:

- Standard shapers such as ZV, UMZV, ZVD, and 2-Mode ZV.
- Radial assisted shapers that use a combination of slewing and radial motions.
- Shapers developed based on the linearized model.

The first set of shapers are based on standard linear shaping theory. The second two sets are special shapers developed specifically for tower cranes.

Figure 7.6 shows the generic process of forming these shaped commands. The figure also introduces some of the terminology and symbols that will be used throughout the chapter. The desired command is assumed to be a slewing velocity pulse. The pulse attains a maximum value of $\dot{s}_f$ and the pulse duration is t_p. The desired command is sent to the command generator, that forms the shaped command. The shaped command begins with a "rising" portion that ends when the command reaches a steady-state value of $\dot{s}_f$. Note that this steady-state velocity identical to the maximum value of the desired command. At time $t = t_p$ the "falling" portion of the shaped command begins. This lasts until the command returns to steady state at rest. For the case of shaped commands that include radial motion, the "rising" and "falling" motions extend until the radial command reaches stead state as well. The figure also shows that the shaped command is not, in general, symmetric about its midpoint.

The following subsections discuss each of these shapers and their limitations. First, a description and derivation of all the shapers will be given, along with their advantages

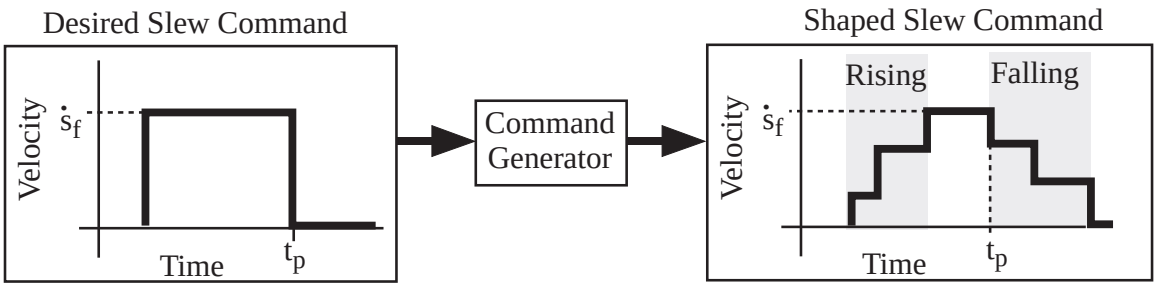


Figure 7.6: Generic Shaping Process

and disadvantages. Then, all of the shapers will be compared in simulation by measuring their performance across various system and motion parameters. Finally, the shapers were experimentally tested on the tower crane and the results will be discussed.

7.3.1 Standard Shapers: ZV, UMZV, and ZVD

Based on the linearized analysis presented in the previous section, ZV, UMZV, and ZVD shapers should work well on tower cranes. If it is assumed that the slew velocity is small, then the system modes are approximately equal to the pendulum mode, as shown in Figure 7.4. Therefore, using a standard ZV or UMZV shaper with a design frequency of $\sqrt{g/L}$ should work well. The unity-magnitude shaper is used because it has the added benefit of being faster than a ZV command.

As the slew velocity is increased the system modes will deviate from $\sqrt{g/L}$ in a linear manner. This can be viewed as a frequency modeling error, which motivates using a ZVD shaper. This shaper should work particularly well, since linearized frequencies are centered about the pendulum frequency. Furthermore, previous studies have shown that robust shapers, such as ZVD, work well on several nonlinear systems [15, 45].

These three shapers have several advantages over the other shapers that will be discussed. They have closed form solutions and are easy to implement. Furthermore, they make no assumptions about the command length, slewing velocity, or radial position.

7.3.2 2 Mode ZV Shaper

Given a slewing velocity, (7.27) can be used to predict the linearized natural frequencies. It would seem logical to use an input shaper that is tuned to these two frequencies, such as

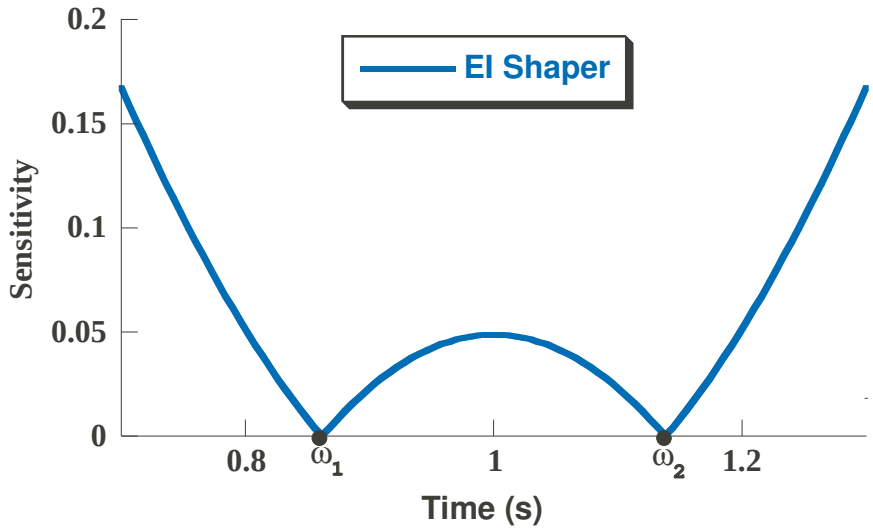


Figure 7.7: Sensitivity Curve for EI Shaper.

a 2-mode ZV shaper (ZV2). Figure 7.7 shows the sensitivity curve of an EI shaper using a design frequency of 1 Hz. Notice that the vibration is zero at ω_1 and ω_2. In addition, the curve is symmetric about the modeled frequency [85]. A ZV2 shaper works in exactly the same way. Based on this figure, the objective is to design the ZV2 shaper so that ω_1 and ω_2 match the linearized natural frequencies of the tower crane. To achieve this goal, the relationship between the frequencies $[\omega_1, \omega_2]$ and the shaper times and amplitudes must be derived.

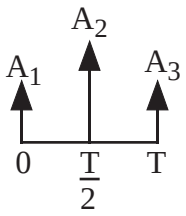
Figure 7.8(a) shows a template for an ZV2 shaper. Note that the shaper times are already determined and $T = 2\pi/\omega_d$ where ω_d is the design frequency. In addition, it will be assumed that:

$$A_1 = A_3 \tag{7.32}$$

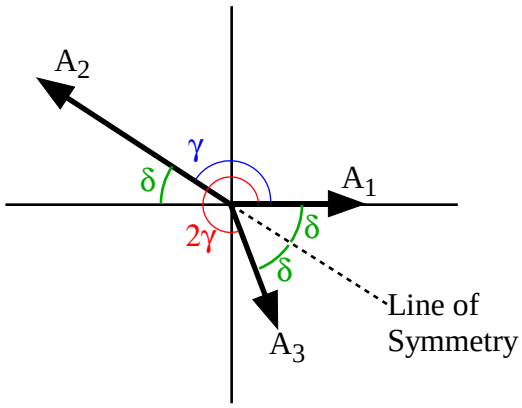
$$A_2 = 1 - 2A_1 \tag{7.33}$$

Assume for the moment that the system frequency is equal to ω_1. The vector diagram for the shaper under these conditions is drawn in Figure 7.8(b). The angle of the second impulse is given by:

$$\gamma = \frac{T\omega_1}{2} = \frac{\pi\omega_1}{\omega_d} \tag{7.34}$$



(a) ZV2 Impulse Times and Amplitudes.



(b) ZV2 Vector Diagram for ω_1.

Figure 7.8: ZV2 Shaper Impulses and Vector Diagram.

where ω_d is the design frequency. The angle of the third impulse, 2γ, is double the angle of the second impulse due to the selection of the shaper times.

Because the vibration is zero at ω_1, the vectors must sum to zero. Given that $A_1 = A_3$, this implies that a line of symmetry exists, collinear with A_2 (shown as the dotted line). A new angle, δ, can be defined as the angle between this line of symmetry and vector A_1. Using basic geometry, δ can be related to other angles in the figure:

$$\delta = \pi \left(1 - \frac{\omega_1}{\omega_d}\right) \tag{7.35}$$

In addition, consider the projection of A_1 or A_3 onto the line of symmetry: $A_1 \cos\delta$. For the vectors to sum to zero, the length of A_2 must equal the sum of these two projections:

$$A_2 = 2A_1 \cos\delta \tag{7.36}$$

A similar vector diagram could be constructed for a system frequency of ω_2. However, because of the pre-selected impulse times and amplitudes the plot would be identical to 7.8(b), but reflected across the x-axis. As a result of this symmetry:

$$\omega_2 - \omega_d = \omega_d - \omega_1 \tag{7.37}$$

The previous equations, (7.32)-(7.37), can be combined to give the relationship between impulse amplitudes and ω_1, ω_2:

$$A_1 = A_3 = \frac{1}{2(1+\alpha)} \qquad\qquad A_3 = 1 - 2A_1 \tag{7.38}$$

$$\alpha = \cos \pi \left(\frac{\omega_2 - \omega_1}{\omega_1} \right)$$

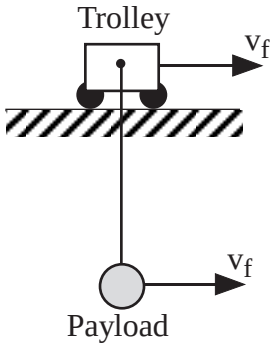
When implementing the ZV2 shaper on the tower crane the question is: *what frequencies should the shaper be designed to cancel?* At first, one might use the linearized frequencies at the final slewing velocity, $\dot{s}_f$. However, the crane never achieves this final velocity until the end of the shaper. Therefore, the first and second impulses induce vibration into the system when the frequencies do not correspond to the linearized frequencies. Instead, the shaper will be tuned for the frequencies at $\frac{1}{2}\dot{s}_f$. This is a first order approximation for the linearized frequencies while the system ramps up to full slewing speed.

7.3.3 Radial Assisted Shapers

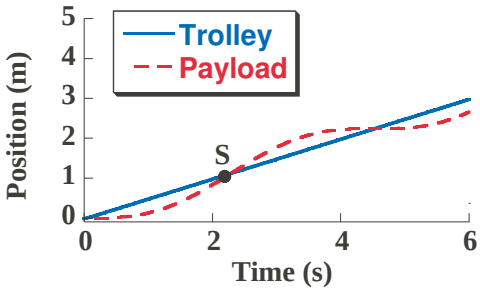
The radial assisted shaper uses a radial motion, in addition to a slewing motion, to reduce the vibration. The shaper is designed to force the system into its steady-state condition while assuming the ϕ and θ states are uncoupled. It is similar to the numerical $\mathrm{ZV}_{\mathrm{fly2}}$ shaper presented in Chapter 6. The radial assisted shaper builds on the relationship between input shaping and the system steady state. First, this relationship will be established for a well-documented problem, ZV shaping a planar crane. Then, these ideas will be carried over and applied to the tower crane.

Consider the simple problem of ZV shaping constant-velocity motion of a planar crane. A ZV shaper moves the planar crane without residual vibration, while the system is moving and when the system stops. In order for the system to be moving without residual vibration, it must be at its steady-state condition for a constant velocity. Figure 7.9(a) shows what this steady-state condition is for the planar crane. The trolley and payload move at constant, and equal velocity v_f. In addition, the payload deflection is zero.

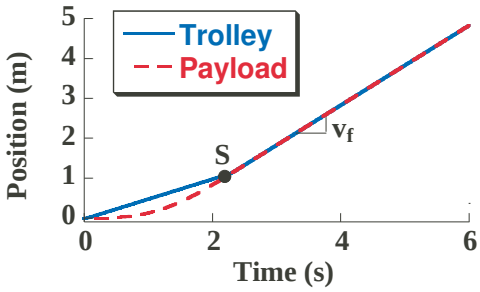
A ZV shaper must bring the system from rest to the steady-state condition shown in Figure 7.9(a). A ZV shaped velocity command consists of two steps; the first step is to half-speed, and the second is to full speed. Figure 7.9(b) shows the planar crane response to a step command to half-speed. The solid line is the trolley position, and the dashed line is the payload position. When the payload reaches the point labeled "S" its velocity is equal to the full-speed velocity of the trolley and the deflection is zero. Recall that this



(a) Steady State



(b) Unshaped Response at Half Speed



(c) ZV Response

Figure 7.9: Relating the Steady State of the Crane to ZV shaping.

was the steady-state condition of the payload given earlier. Therefore, if the trolley speed were to suddenly increase to full-speed at point "S", the system would be at steady state and there would be no vibration. Figure 7.9(c) shows the full ZV shaped command and response. As predicted, the trolley velocity switches to full on at precisely the point "S" when the payload is at the steady-state condition.

Based on the previous observations, a ZV shaper can be interpreted to work as follows: the shaper moves the system initially, waits for the system to reach a critical point, then changes so the system "relaxes" to steady-state. This idea is very old, and has its roots in the original "Posi-cast" control developed by Smith in the late 1950s [94]. A similar approach was used by Smith, et. al. for a system with a linearly-varying natural frequency [92]. The term "steady-state relaxation" will be used to refer to this strategy in this paper.

The same "steady-state relaxation" idea can be applied to the tower crane system, with a few modifications. First, the equilibrium condition of the tower crane for a constant

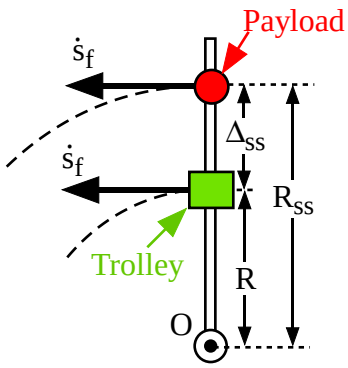


Figure 7.10: Tower Crane Steady State Condition.

slewing velocity must be established. This problem was solved in the previous section: at steady state the payload is radially swung outward, but remains in the plane of the mast and jib. Figure 7.10 shows an overhead view of what this steady-state condition looks like. Both the trolley and payload have the same instantaneous angular velocity, $\dot{s}_f$. The steady-state payload deflection, Δ_{ss}, is related to the steady-state angle, ϕ_{ss}, given in (7.21) by: $\Delta_{ss} = L \sin \phi_{ss}$. The steady-state radial position of the payload, R_{ss}, is therefore given by:

$$\begin{aligned} R_{ss} &= R + \Delta ss \\ &= R + L \sin \phi_{ss} \end{aligned} \tag{7.39}$$

where R is the radial position of the trolley.

Forming the radial assist command is a three stage process, as shown in Figures 7.11-7.13. In the first stage, a ZV-shaped slewing command is used. Figure 7.11 shows the simulated response in polar coordinates. These polar coordinates were previously defined in Section 7.1. Figure 7.11(a) shows the trolley and payload angular position. Figure 7.11(b) shows the trolley and payload radial position along with the instantaneous steady-state radial position, R_{ss}. The ZV shaper appears to work well for the angular response and resembles the response of the planar crane shown earlier. However, the ZV command has induced vibration in the radial direction. When the slewing velocity reaches full speed, R_{ss} jumps to a new position. As a result the payload is no longer at steady state and oscillates about this new steady-state value. The whole purpose of this simulation is to identify the time and value of the first peak in the radial response, labeled P in the figure. This information will be used in the second stage to eliminate the vibration.

The second stage of forming the radial assist command is to eliminate the residual vibration using the same "steady-state relaxation" ideas presented earlier. In this case the first peak, labeled P, plays the role of the steady-state point, S, shown earlier for the planar crane. Point P satisfies the steady-state condition of zero velocity, but the position does not equal R_{ss}. However, R_{ss} can be adjusted by moving the trolley. More specifically, (7.22) can predict the trolley radial position that makes R_{ss} equal to P. The trolley is then commanded to quickly move to this new position at the same time as point P. Figure 7.12 shows the resulting response. The trolley no longer oscillates because it is at the steady-state radius, R_{ss}.

The above command only brings the system from rest to a constant slew-velocity. In the third stage of forming the radial assist command the above process is repeated so the system returns to rest. The total command is shown in Figure 7.13. The system was commanded to stop at $t = 3s$. Note that the command is *not* symmetric. Also note that the command length does not need to be known ahead of time.

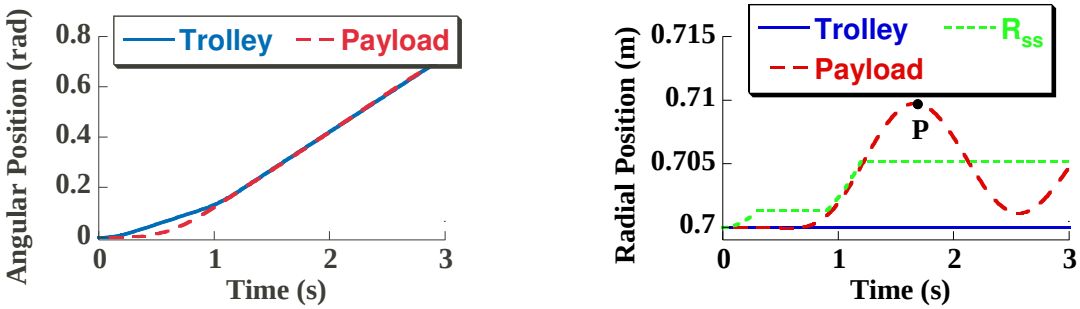
7.3.3.1 Advantages and Disadvantages of Radial Assisted Shapers

There are several advantages and disadvantages to the above technique. The advantages are:

- It captures many of the non-linear aspects of the model.
- The technique is guaranteed to do better then ZV shaping alone.
- The technique can be easily expanded to form different types of commands. In this case a ZV shaper was used for the slewing velocity, but any other shaper could also be used.

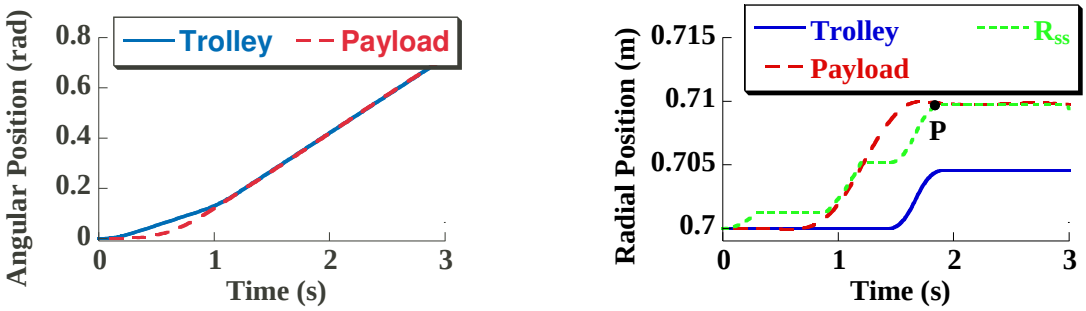
The disadvantages are:

- Simulating the system can be computationally expensive and new simulations must be run for any change in parameters.
- The process assumes that the slewing and radial vibration are essentially uncoupled.



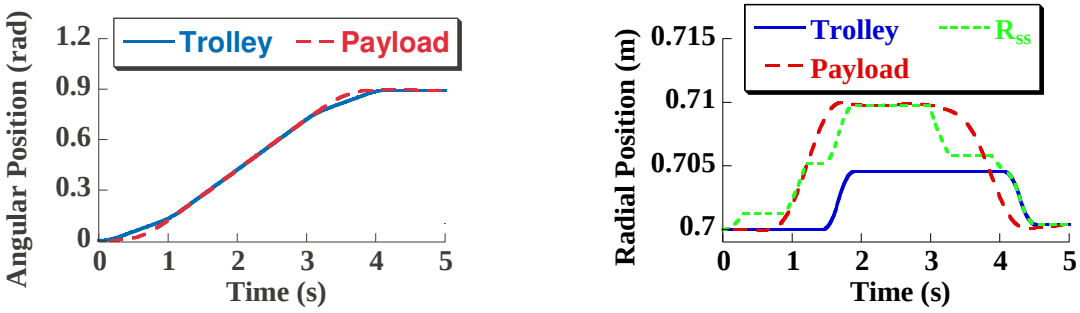
(a) Angular Position (b) Angular Position

Figure 7.11: Radial Assist Shaper Formation: First Stage.



(a) Angular Position (b) Radial Position

Figure 7.12: Radial Assist Shaper Formation: Second Stage.



(a) Slew Position (b) Radial Position

Figure 7.13: Radial Assist Shaper Formation: Third Stage.

It has already been shown that this is not the case, which is why this technique will always have some small residual vibration.

- Because the command is not symmetric, the final radial position is never exactly the same as the initial radial position. However, in all the simulations this difference was very small. The maximum measured change between the initial and final trolley radius between was 2% for the crane parameter ranges: $R = [0.5 \ldots 1]$ m, $L = [0.5 \ldots 1.5]$ m, and $\dot{s} = [0.1 \ldots 0.5]$ rad/s. It was observed that the percent change in trolley radius increased with the suspension length, L, and slew velocity, $\dot{s}$. However, the trolley radius, R, had a negligible effect on the percent change in trolley radius.

- This technique is only applicable for sufficiently long commands. For example, referring to Figure 7.12(b), the system cannot be commanded to stop before the time of point P. In other words t_p cannot be less than the time of point P. Point P represents the time when the "rising" portion of the shaped command is complete and the system has reached steady state. This rule applies for any command generated with this method. If the desired command length violates this constraint, a different shaper would need to be employed (eg. ZVD).

As mentioned above, this technique is very flexible. The three-stage process discussed earlier can be generalized as follows:

1. Simulate the system using a shaped step command for the slewing axis, where the shaper can be any linear shaper (eg. ZV, UMZV, ZVD, or other).

2. Measure the time and value of the first peak of the radial swing. Then, generate a step in the trolley radial position such that the steady-state payload radius, R_{ss}, coincides with the value first peak at the measured time.

3. Repeat the process to generate the command that returns the system to rest.

This thesis will examine two different cases: a ZV based command (as illustrated above), called ZV_R, and a UMZV based command, called UMZV_R. Both were formed using the three step process outlined above.

7.3.4 Linearized Shapers

This section develops a shaped command for the slewing axis (no radial motion) that is designed around the linearized model developed in Section 7.2. The general strategy is to examine how a multi-step command induces vibration in the z-coordinate system, defined in (7.30), and then, choose a command that eliminates the vibration. The z-coordinate system is used because the modes are uncoupled, making it easier to solve for zero-vibration commands. Using these ideas, deriving the shaped command is a three step process:

1. Examine how a staircase slewing command induces vibration in the z-coordinates. More specifically, given an arbitrary step change at time t_i with amplitude A_i, what will be the change in the magnitude and phase of vibration in the z-coordinates?

2. Analytically solve for the "rising" portion of the command that will induce zero vibration in the first mode, defined by the z_1 and z_2 coordinates.

3. While satisfying step 2, above, numerically solve for the "rising" portion of the command that minimizes vibration in the second mode, defined by z_3 and z_4.

4. Repeat the process for the "falling" portion of the command.

This process is carried out using two different shaper templates: a 2-mode ZV shaper, and a 2-mode UMZV shaper.

7.3.4.1 Relationship between Step Commands and Z-Coordinates

The analysis of the linearized equations of motion can be summarized as follows:

- A linearized model of the system is given by (7.25) using the physical coordinates $[\theta\ \dot{\theta}\ \bar{\phi}\ \dot{\phi}]^T$.

- Using the transformation, T, given by (7.28) and (7.30) the physical coordinates are transformed into $[z_1 z_2 z_3 z_4]^T$.

- This transformation isolates the two modes of the system and brings the system into Jordan form (7.31). The first mode is contained in z_1, z_2 and the second is contained in z_3, z_4.

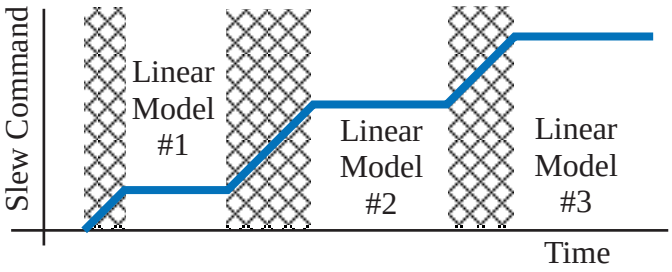


Figure 7.14: Limitations of Linear Model.

Consider how this linearized model can be used to predict the response to a general three-step command. Figure 7.14 shows an arbitrary slewing command. The hatched areas are times where the linear model does not apply because the slewing velocity is changing. During the constant velocity portions of the command a different linear model must be used for each step. This is because the steady-state slewing velocity, $\dot{s}_{ss}$, changes the linear model.

The system response during the constant velocity portions of the command are easy to derive because the linear model can be used. What is unknown, at this point, is how the response is effected by the velocity changes during the hatched regions. This determines the initial conditions for the linear model during the next constant velocity segment. Therefore, the first question to be addressed is: *how do the velocity changes effect the response?*

Assume, for the moment, that the velocity changes are instantaneous. Using this assumption it follows that:

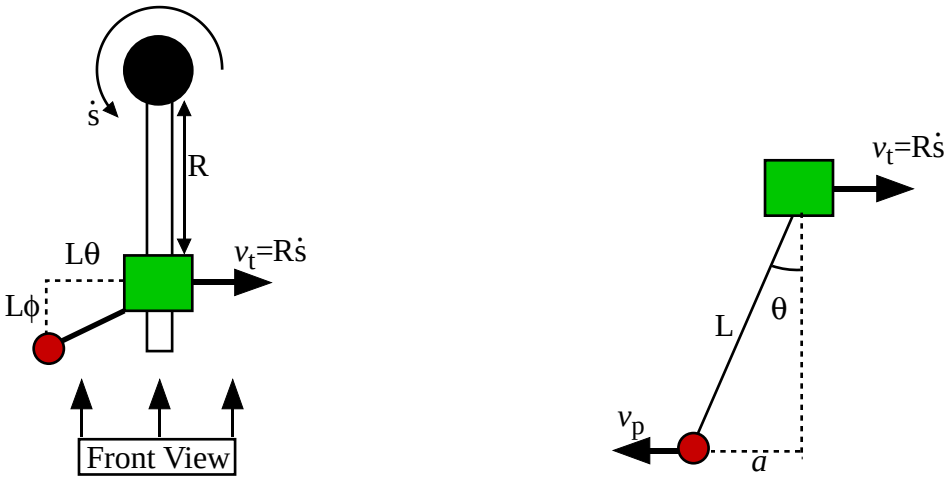
The absolute position of the payload remains fixed during the velocity changes.

This assumption is the core concept for deriving the new linearized shaper. However, velocity changes of a real system can never be instantaneous. But if the velocity changes rapidly relative to the system dynamics, the above assumption is still a good approximation. Therefore, the assumption can be modified slightly to read:

The absolute position of the payload remains fixed during rapid velocity changes.

(7.40)

Using the above assumption and the crane geometry, one can determine the effect of a rapid velocity change on the ϕ, θ-coordinates.



(a) Top View (b) Front View

Figure 7.15: Physical Configuration during Step Transition.

Figure 7.15 shows the crane geometry at an arbitrary time using a top view (a) and a frontal view (b). For small deflection angles the coordinates of the payload, relative to the trolley, can be approximated by $L\phi$ and $L\theta$, as shown in Figure 7.15(a). As a result of assumption (7.40), it follows that ϕ and θ will remain unchanged during rapid changes in rotational velocity. In addition, $\dot{\phi}$ will remain unchanged because trolley velocity is zero in the $\dot{\phi}$-direction. However, $\dot{\theta}$ will change. Using the top view 7.15(a) the instantaneous trolley velocity is given by:

$$v_t = R\dot{s} \tag{7.41}$$

Using the frontal view shown in figure 7.15(b), the angle θ can be related to the distance, a:

$$a \approx L\theta \tag{7.42}$$

assuming θ is small. The distance, a, can then be related to the trolley velocity, v_t, and payload velocity, v_p:

$$\begin{aligned} \dot{a}_1 &= v_{t1} - v_p \\ \dot{a}_2 &= v_{t2} - v_p \end{aligned} \tag{7.43}$$

where $\dot{a}_1$ and $\dot{a}_2$ are the values of $\dot{a}$ for the initial and final slew velocities, and v_{t1} and v_{t2} are the values of v_t for the initial and final slew velocities. The change in $\dot{a}$ is then given

by:

$$\begin{aligned} \Delta\dot{a} &= \dot{a}_2 - \dot{a}_1 \\ \Delta\dot{a} &= v_{t2} - v_{t1} \\ \Delta\dot{a} &= R(\dot{s}_2 - \dot{s}_1) \end{aligned} \tag{7.44}$$

where $\Delta\dot{a}$ is the change in $\dot{a}$, $\dot{s}_1$ is the initial slew velocity, and $\dot{s}_2$ is the final slew velocity. Finally, (7.44) can be combined with (7.42) to yield:

$$\Delta\dot{\theta} = \frac{R}{L}(\dot{s}_2 - \dot{s}_1) \tag{7.45}$$

To summarize:

$$\begin{aligned} \Delta\theta &= 0 \\ \Delta\dot{\theta} &= \frac{R}{L}(\dot{s}_2 - \dot{s}_1) \\ \Delta\phi &= 0 \\ \Delta\dot{\phi} &= 0 \end{aligned} \tag{7.46}$$

where $\Delta\theta, \Delta\dot{\theta}, \Delta\phi$, and $\Delta\dot{\phi}$ give the changes in the states $\theta, \dot{\theta}, \phi$, and $\dot{\phi}$ due to the rapid change in slewing velocity $(\dot{s}_2 - \dot{s}_1)$.

The next step is to convert the change of physical coordinates, given in (7.46), to a change in the linearized z-coordinates. More precisely the question is: given some initial state in z-coordinates, $\vec{z}_1$, what will be the final state, $\vec{z}_2$, due to the rapid change in slew velocity. Mathematically, the answer is found by transforming $\vec{z}_1$ to physical coordinates, adding the change in physical coordinates given in (7.46), and then transforming back into z-coordinates to yield $\vec{z}_2$. The result is:

$$z_2 = T_2^{-1}T_1 + T_2^{-1}\begin{bmatrix} 0 \\ \frac{R}{L}(\dot{s}_2 - \dot{s}_1) \\ \phi_{ss1} - \phi_{ss2} \\ 0 \end{bmatrix} \tag{7.47}$$

where T_2 and T_1 are the transformation matrices found by substituting the slewing velocities $\dot{s}_2, \dot{s}_1$ into (7.28). Also note that ϕ_{ss1}, ϕ_{ss2} are the linearized equilibrium ϕ-angles given by (7.22) for slewing velocities $\dot{s}_2, \dot{s}_1$ respectively.

Equation (7.47) can be simplified by examining the $T_2^{-1}T_1$ term. Expanding this term out yields:

$$T_2^{-1}T_1 = \begin{bmatrix} 1+\frac{\alpha}{2} & 0 & -\frac{\alpha}{2} & 0 \\ 0 & 1+\frac{\alpha}{2} & 0 & \frac{\alpha}{2} \\ \frac{\alpha}{2} & 0 & 1-\frac{\alpha}{2} & 0 \\ 0 & -\frac{\alpha}{2} & 0 & 1-\frac{\alpha}{2} \end{bmatrix}, \qquad \alpha = \frac{\Delta\omega_1}{\omega_0} \tag{7.48}$$

where $\omega_0 = \sqrt{g/L}$ is the pendulum frequency of the crane, and $\Delta\omega_1$ is the change in the first, linearized frequency due to the change in slew velocity. Note that α could have also been defined as, $\alpha = -\frac{\Delta\omega_2}{\omega_0}$, where $\Delta\omega_2$ is the change in the second, linearized frequency. The non-dimensional term, α represents the change in the linearized frequencies (either ω_1 or ω_2) relative to the pendulum frequency. For small slewing velocities, this term will be small, $\alpha \ll 1$. Therefore, it follows that $T_2^{-1}T_1$ can be approximated by an identity matrix for small slewing velocities. Substituting this assumption into (7.47), along with the formulas for $\phi_{ss1}, \phi_{ss2}, T_2^{-1}$, and simplifying yields:

$$\Delta z = \begin{bmatrix} 0 \\ -\frac{R}{2L\omega_0}\left(\frac{\dot{s}_2\omega_0}{\omega_3} + \dot{s}_1\left(-1+\frac{\dot{s}_1\omega_4}{\omega_1\omega_2}\right)\right) \\ 0 \\ \frac{R}{2L\omega_0}\left(\frac{\dot{s}_2\omega_0}{\omega_4} - \dot{s}_1\left(1+\frac{\dot{s}_1\omega_3}{\omega_1\omega_2}\right)\right) \end{bmatrix}, \qquad \begin{aligned} \omega_1 &= \omega_0 + \dot{s}_1 \\ \omega_2 &= \omega_0 - \dot{s}_1 \\ \omega_3 &= \omega_0 + \dot{s}_2 \\ \omega_4 &= \omega_0 - \dot{s}_2 \end{aligned} \tag{7.49}$$

Equation (7.49) gives the relationship between a rapid change of slewing velocity, and the change in the linearized z-coordinates.

A multi-step response was simulated to test the validity of these results. Figure 7.16 shows the simulation results. The top plot, Figure 7.16(a), shows the 2-step slewing command for the tower crane. Figures 7.16(b) and 7.16(c) show the response using the physical coordinates $[\phi,\ \dot{\phi},\ \theta,\ \dot{\theta}]$. Notice that when the command changes to a new velocity, at 0 seconds and 3 seconds, all of the physical coordinates change very little except for $\dot{\theta}$. This was already predicted by (7.46). Figures 7.16(d) and 7.16(e) show the response in the z-coordinate system. When the command changes to a new velocity, at 0 seconds and 3 seconds, states z_1, z_3 change very little, while states z_2, z_4 rapidly change. Again, this was predicted by (7.49).

It should be emphasized that although the above derivation calculated the instantaneous change for the states $\dot{\theta}, z_1$, and z_3, these values did not literally change instantaneously in the simulation. The same would hold for a real system. The assumption that these states change instantaneously is only a convenient approximation for deriving the magnitude of their change.

The issue of how a change in slew velocity effects the z-response has been addressed. What remains is to determine how the system responds during the constant velocity portions of a slewing command, as previously shown in Figure 7.14. During these portions of the response, the system can be well approximated using the linearized model (7.31). The first mode is contained in z_1 and z_2 and the second is contained in z_3 and z_4. At this point, it will be useful to define the complex response of each mode as:

$$\begin{aligned} v(t) &= z_2(t) + z_1(t)i \\ y(t) &= z_4(t) + z_3(t)i \end{aligned} \tag{7.50}$$

where $v(t)$ is the complex response of the first mode, and $y(t)$ is the complex response of the second mode. Because the linearized system (7.31) is in Jordan form the free response to any given initial conditions, v_0, y_0, is:

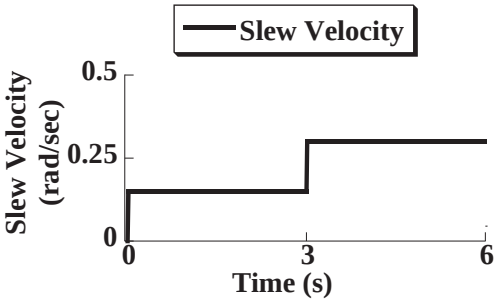
$$v(t) = v_0 e^{i\omega_1 t}, \qquad \omega_1 = \omega_n + \dot{s} \tag{7.51}$$

$$y(t) = w_0 e^{i\omega_2 t}, \qquad \omega_2 = \omega_n - \dot{s} \tag{7.52}$$

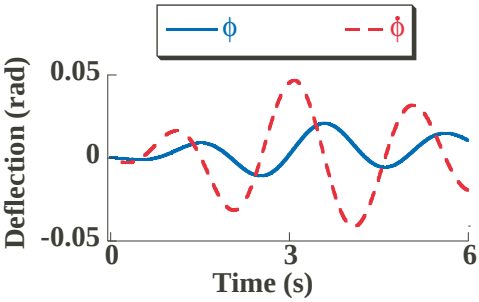
The z-response to a staircase command has now been completely described. As a final refinement, the result in (7.49) can be written in terms of the complex coordinates (7.50):

$$\Delta v_{jk} = -\frac{R}{2L\omega_0}\left(\frac{\dot{s}_k\omega_0}{\omega_3} + \dot{s}_j\left(-1 + \frac{\dot{s}_j\omega_4}{\omega_1\omega_2}\right)\right) \tag{7.53}$$

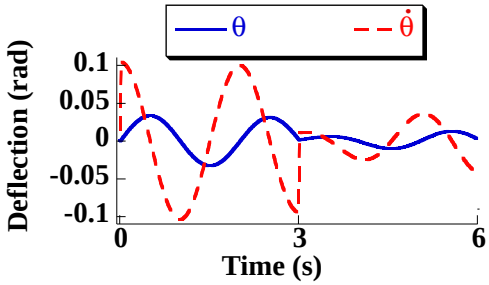
$$\Delta y_{jk} = \frac{R}{2L\omega_0}\left(\frac{\dot{s}_k\omega_0}{\omega_4} - \dot{s}_j\left(1 + \frac{\dot{s}_j\omega_3}{\omega_1\omega_2}\right)\right) \tag{7.54}$$



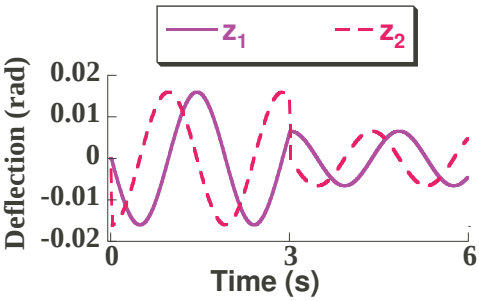
(a) Command



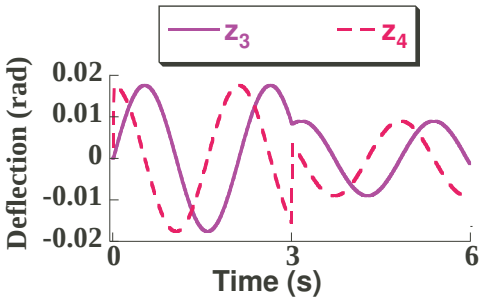
(b) ϕ and $\dot{\phi}$



(c) θ and $\dot{\theta}$



(d) z_1 and z_2



(e) z_3 and z_4

Figure 7.16: Response to Multi-Step Slew Command.

where,

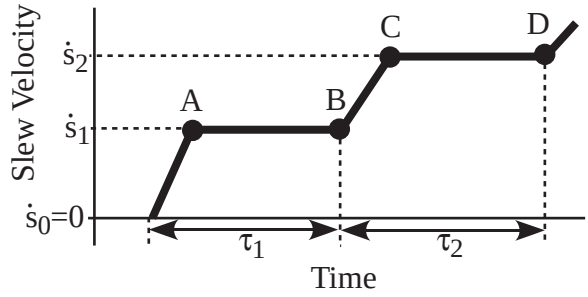
$$\begin{aligned}\omega_1 &= \omega_0 + \dot{s}_j \\ \omega_2 &= \omega_0 - \dot{s}_j \\ \omega_3 &= \omega_0 + \dot{s}_k \\ \omega_4 &= \omega_0 - \dot{s}_k\end{aligned}$$

Δv_{jk} and Δy_{jk} refer to the change in v and y due to a step change in slewing velocity from $\dot{s}_j$ to $\dot{s}_k$ and $\omega_0 = \sqrt{g/L}$. Notice that in both cases the change in the complex response is real valued.

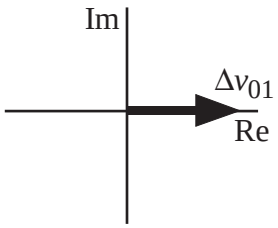
Now, the response to a staircase command can be found by alternating application of (7.51), (7.52) and (7.53), (7.54). Figure 7.17 gives an example of how to construct the response using vector diagrams. Consider the command shown in Figure 7.17(a). The critical points for analysis are before and after each velocity change. These points are labeled A-D. Note that the point before the first velocity change is not labeled because it is the trivial case where all the states are zero. The velocities after each transition are labeled $\dot{s}_1$ and $\dot{s}_2$ respectively, while the initial velocity of zero is labeled $\dot{s}_0$.

Point A The first transition, to point A, is from zero velocity to $\dot{s}_1$. The change in the complex response of the first mode, Δv_{01}, is determined by substituting $\dot{s}_0$ and $\dot{s}_1$ into (7.53). In the complex plane this effect is plotted as a horizontal arrow to the right, as shown in Figure 7.17(b). The process is repeated using (7.54) to determine the change in y. The results are plotted in the complex plane as shown in Figure 7.17(c).

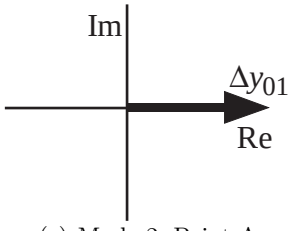
Point B During the transition from A to B the slewing velocity is constant. According to (7.51) the first-mode vector, v, will rotate about an angle $\omega_1\tau_1$. τ_1 is the time duration between points A and B and ω_1 is the linearized frequency of the first mode given by $\omega_1 = \omega_0 + \dot{s}_1$. This rotation is shown in Figure 7.17(d). Similarly, (7.52) gives the rotation of y as $\omega_2\tau_1$, as shown in Figure 7.17(e). ω_2 is the linearized frequency of the second mode given by $\omega_2 = \omega_0 - \dot{s}_1$.



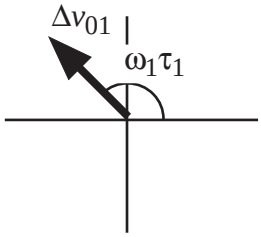
(a) Command



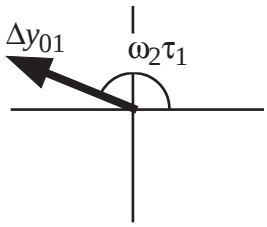
(b) Mode 1, Point A



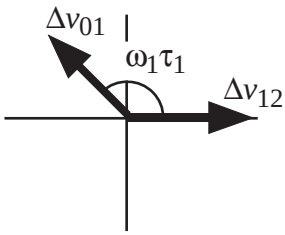
(c) Mode 2, Point A



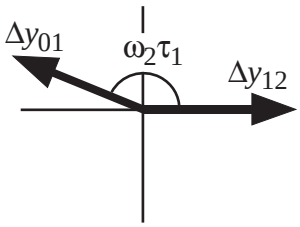
(d) Mode 1, Point B



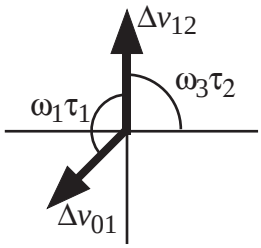
(e) Mode 2, Point B



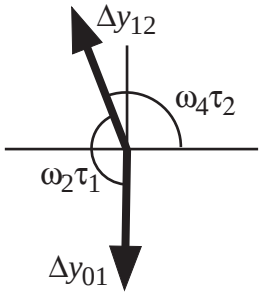
(f) Mode 1, Point C



(g) Mode 2, Point C



(h) Mode 1, Point D



(i) Mode 2, Point D

Figure 7.17: Forming Vector Diagram to Multi-Step Slew Command.

Point C During transition from B to C the slewing velocity changes from $\dot{s}_1$ to $\dot{s}_2$. So once again, (7.53) can be used to determine Δv_{12}. This is plotted along the real axis in Figure 7.17(f). The total vibration is therefore given as the vector sum of the two vectors shown in the figure. Similarly, (7.54) is used to determine the change in y Δy_{12}, which is plotted in 7.17(g).

Point D During the transition from C to D the slewing velocity is constant. According to (7.51), the first-mode vectors will be rotated about an angle $\omega_3 \tau_2$. τ_2 is the time duration between points C and D and ω_3 is the linearized frequency of the first mode given by $\omega_3 = \omega_0 + \dot{s}_2$. Notice in Figure 7.17(h) that both vectors are rotated by the same angle. The same procedure is carried out on the second mode vector, y, as shown in Figure 7.17(i). In this case, the vectors are rotated by $\omega_4 \tau_2$ where ω_4 is the linearized frequency of the second-mode given by $\omega_4 = \omega_0 - \dot{s}_2$.

The above process can be continued for any staircase-type command to yield a good approximation for the response. To extract the values of the z-coordinates from the vector diagram; all the vectors in the diagram are summed and the real and imaginary parts are evaluated to yield $[z_1 \cdots z_4]$ as in (7.50). The process is very similar to the vector-diagram approach for linear systems [85].

One important observation should be made here. According to (7.53) and (7.54), the trolley radial distance, R, has only one effect on the vibration of the first and second mode: it linearly scales the amplitude. This means that given any arbitrary staircase command, the vibration amplitude of the two modes at any point in time is linearly dependant on R. The phase is unaffected by the value of R. In the next sections of this chapter, input shapers will be derived by setting the total amplitude of each mode equal to zero. As a result, these input shapers will only depend on the final slew velocity, $\dot{s}_f$, and the suspension length, L. Because the trolley radius merely scales the vibration, it is not a factor for designing these types of shapers.

The first stage in the development of linearized input shapers is complete: a simple method for predicting the amplitude and phase of vibration in z-space for an arbitrary

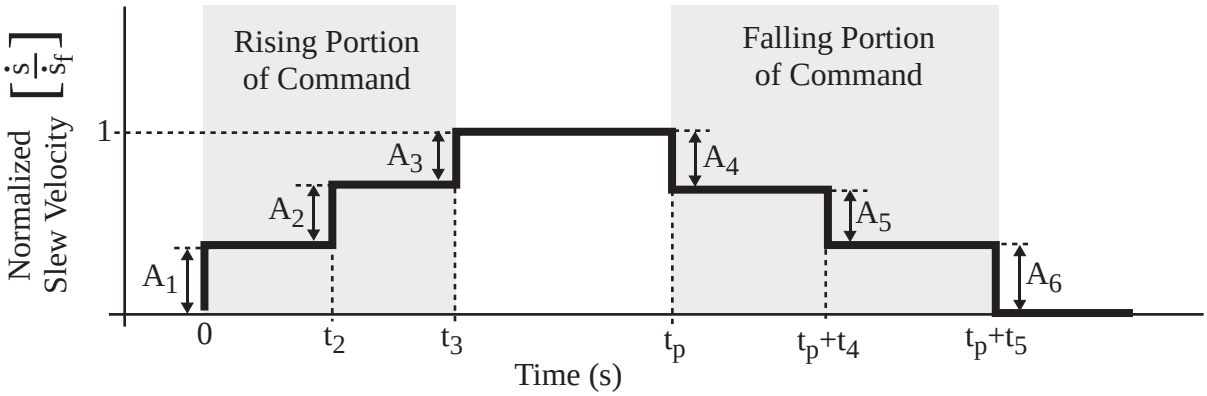


Figure 7.18: $ZV2_{Lin}$ Command Template.

staircase command was developed. The next stage is to derive input shapers by minimizing the total vibration at the end of the command. This process was carried out using two different shaping templates: a 2-mode ZV template and a 2-mode UMZV template. As mentioned at the beginning of this section, the solution process for each begins by finding an analytical solution that eliminates residual vibration in the first mode. Then, the residual vibration in the second mode is numerically minimized, while constraining the residual vibration of first mode to remain at zero.

7.3.4.2 Linearized, 2-Mode ZV Command

A linearized, 2-mode, ZV command, called $ZV2_{Lin}$, is developed in this section. A template for the command is shown in Figure 7.18. Note that the vertical axis is normalized by the final setpoint slew velocity, $\dot{s}_f$. The step amplitudes, A_i, are normalized so $A_1+A_2+A_3 = 1$ and $A_4 + A_5 + A_6 = -1$. Similar to a linear, 2-mode ZV shaper, the command is a three-step staircase. However, the "rising" portion of the command is different from the "falling" portion (ie. the command is not symmetric). A three-step staircase is assumed because the system has two modes of vibration for any non-zero slewing velocity. This is the minimum number of steps needed to cancel two modes in linear systems, and since this technique is based on the linearized model the same idea holds here.

As mentioned earlier, the 2-mode ZV shaper is derived by setting the amplitude of the two modes equal to zero at the end of the staircase command. Using (7.51)-(7.54) this

becomes:

$$0 = \Delta v_{01} e^{i(\omega_1 t_2 + \omega_3(t_3 - t_2))} + \Delta v_{12} e^{i\omega_3(t_3 - t_2)} + \Delta v_{23} \tag{7.55}$$

$$0 = \Delta y_{01} e^{i(\omega_2 t_2 + \omega_4(t_3 - t_2))} + \Delta y_{12} e^{i\omega_4(t_3 - t_2)} + \Delta y_{23} \tag{7.56}$$

where $\Delta v_{jk}, \Delta y_{jk}, \omega_1, \omega_2, \omega_3, \omega_4$ are derived from (7.53) and (7.54) using slew velocities, $\dot{s}_j$. These slew velocities during the rising portion of the command can be found from the step amplitudes A_1 and A_2:

$$\begin{aligned} \dot{s}_0 &= 0 & \dot{s}_1 &= \dot{s}_f A_1 \\ \dot{s}_2 &= \dot{s}_f (A_1 + A_2) & \dot{s}_3 &= \dot{s}_f \end{aligned} \tag{7.57}$$

The times t_2 and t_3 refer to the step times shown in Figure 7.18. Note that when (7.55) is satisfied the residual vibration in the first mode will be zero. Likewise, when (7.56) is satisfied the residual vibration in the second mode will be zero.

To derive the ZV2_{Lin} command, one would need to solve (7.55) and (7.56) for A_1, A_2, t_2, and t_3. Note that there are really four equations since both imaginary and real parts must equal zero. However, the problem can be simplified by analytically solving one of the equations. The first mode equation, (7.55), can be solved for t_2 and t_3 given any A_1 and A_2. Figure 7.19 shows the vector diagram representation of (7.55). The angles α_1 and α_2 are used to simplify the diagram, and can be related back to (7.55) via:

$$\alpha_1 = \omega_1 t_2 + \omega_3(t_3 - t_2)) \qquad \alpha_2 = \omega_3(t_3 - t_2) \tag{7.58}$$

Both vectors Δv_{01} and Δv_{12} have been copied and translated into the first quadrant, as shown by the dashed arrows. This will simplify the derivation.

For the vectors to sum to zero, they must form a closed polygon when aligned head-to-tail. In this case, the vectors formed by Δv_{01}, Δv_{12}, and Δv_{23} must form the triangle shown in the first quadrant. The law of cosines can be employed to relate the leg lengths to angles a_1 and a_2:

$$(\Delta v_{12})^2 = (\Delta v_{01})^2 + (\Delta v_{23})^2 - 2|\Delta v_{01}||\Delta v_{23}| \cos a_1 \tag{7.59}$$

$$(\Delta v_{01})^2 = (\Delta v_{12})^2 + (\Delta v_{23})^2 - 2|\Delta v_{11}||\Delta v_{23}| \cos a_2 \tag{7.60}$$

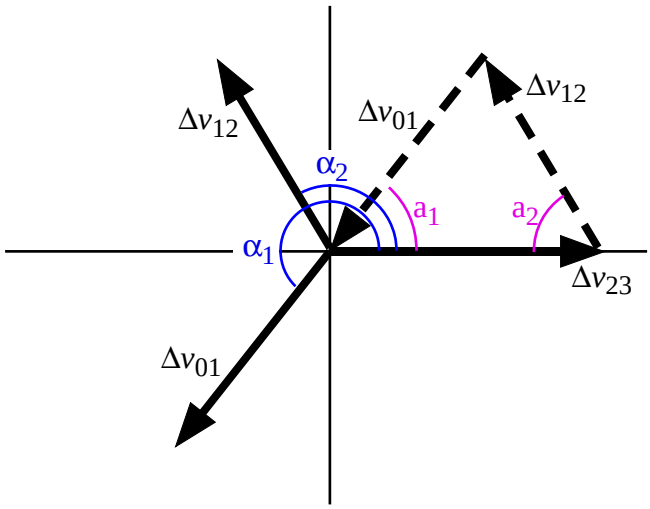


Figure 7.19: Vector Diagram for $ZV2_{Lin}$Slewing Command.

The angles a_1 and a_2 are related to angles α_1 and α_2 via:

$$\alpha_2 = \pi - a_2 \qquad\qquad \alpha_1 = \pi + a_1 \tag{7.61}$$

Given A_1 and A_2 one would solve for $\Delta v_{01}, \Delta v_{12}, \Delta v_{23}$, then substitute into (7.58)-(7.61) to solve for times t_2 and t_3. This ensures that the first mode has zero residual vibration.

To solve for the $ZV2_{Lin}$ command both (7.55) and (7.56) must equal zero. This can be achieved using three steps:

1. An initial guess of $A_1 = 0.25, A_2 = 0.5, A_3 = 0.25$ (similar to a ZVD shaper) was used.

2. The above process was used to find times t_2 and t_3 that zeroed the vibration in the first mode and satisfied (7.55).

3. This formed the initial guess in a nonlinear optimization routine. In this case Matlab's *fmincon* function was employed. The function was programmed to minimize the norm on the right-hand-side of (7.56), while constraining (7.55) to remain equal to zero.

Figure 7.20 shows the solution for A_1 for various slew velocities, $\dot{s}_f$, and suspension lengths, L. Similar plots were constructed for the other shaper parameters. Recall that the trolley radial length, R, does not effect the solution. To simplify the solution, a 3^{rd} degree,

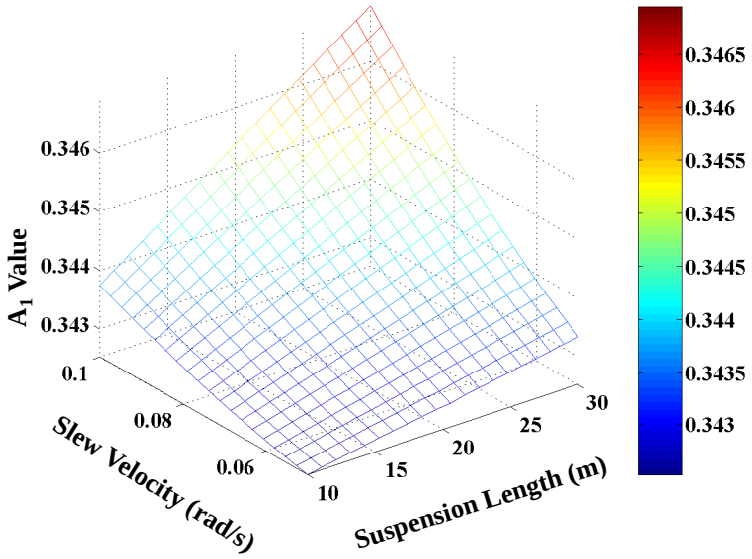
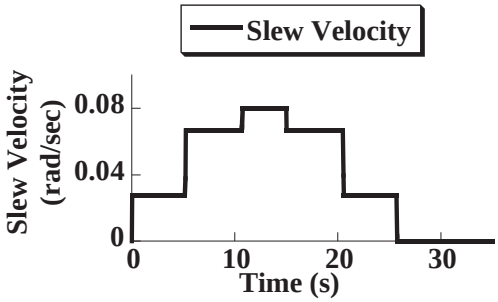


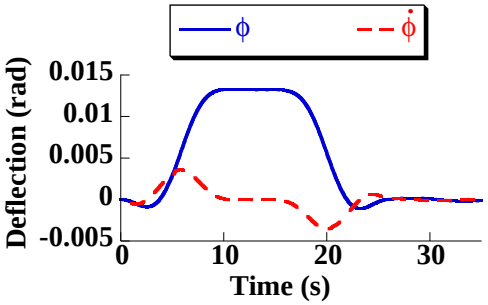
Figure 7.20: A_1 values of $ZV2_{Lin}$ for Various Parameters.

2-dimensional polynomial was fit to each surface. A 3^{rd} degree polynomial was chosen because it was the lowest degree polynomial that yielded acceptable error (1% or less). The polynomial fit was performed for a system parameter range of: $\dot{s}_f = [0.05 \ldots 0.1]$ (rad/sec) and $L = [10 \ldots 30]$ (m). These ranges were chosen based on the system parameters for industrial tower cranes. The polynomial coefficients are reported in Appendix C.2. The entire process was then repeated to find the parameters t_4, t_5, A_4, A_5, and A_6 for the falling portion of the command, as shown in Figure 7.18. To form the shaped command, the suspension length and slew velocities are substituted into the polynomial given in Appendix C.2. The resulting shaper parameters $A_1, \ldots, A_6$ and $t_1, \ldots, t_6$ are then used to construct the command as shown in Figure 7.18.

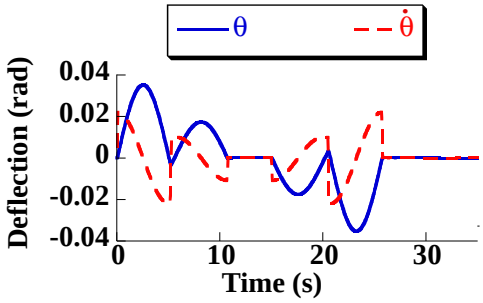
A simulation of the ZV2_{Lin} command and response appears in Figure 7.21. The simulation parameters were: $L = 20$ m, $\dot{s} = 0.08$ rad/s, $R = 20$ m, and $t_p = 15$ s. Figure 7.21(a) shows the ZV2_{Lin} command. Figures 7.21(b) and 7.21(c) show the response of the physical coordinates $\phi, \dot{\phi}, \theta$, and $\dot{\theta}$. Note that the residual vibration is very low during the constant slew velocity segment in the middle of the command, and at the end of the command. The small residual vibration appearing at the end of the command is due to errors that accumulate over the course of the response. These errors come from deviations between



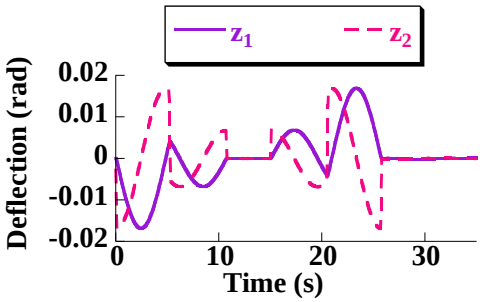
(a) $ZV2_{Lin}$ Command



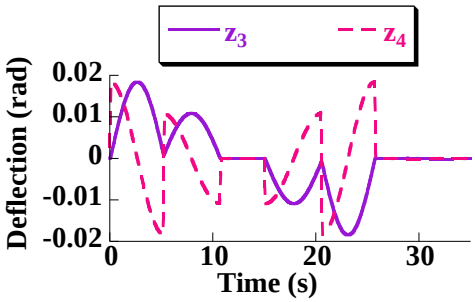
(b) ϕ and $\dot{\phi}$



(c) θ and $\dot{\theta}$



(d) z_1 and z_2



(e) z_3 and z_4

Figure 7.21: $ZV2_{Lin}$ command and response.

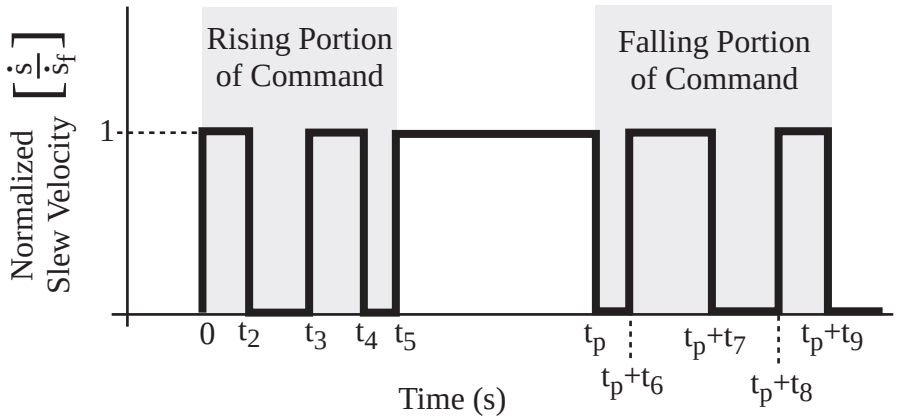


Figure 7.22: UMZV2$_{Lin}$ Shaper Template.

the linearized model and the actual nonlinear model. Figures 7.21(d) and 7.21(e) show the response of the z_1, z_2, z_3, and z_4 coordinates. Note that these values are essentially zero during the constant slew velocity segment in the middle of the command, and at the end of the command. This indicates that the ZV2$_{\text{Lin}}$ command minimizes the vibration in the two linearized modes.

7.3.4.3 Linearized, 2-Mode UMZV Command

In the previous section, a 2-mode ZV command was used as a template for forming a zero vibration command. In this section, a 2-mode UMZV command will be used as the template to form the linearized command, UMZV2$_{\text{Lin}}$. Figure 7.22 shows this template. At any given time the commanded velocity is either zero, or its maximum value, $\dot{s}_f$. Because the velocities are already pre-determined, the unknown shaper parameters are the switch times: $t_2 \ldots t_5$ for the rising portion, and $t_6 \ldots t_9$ for the falling portion.

Because the UMZV command is an on/off command there are only two possible velocity transitions: a positive transition from zero to $\dot{s}_f$, and a negative transition from $\dot{s}_f$ to zero. Therefore, the effects of each transition on v and y can be calculated independent of the switch times using (7.53) and (7.54):

$$\Delta v_p = -\frac{C}{\omega_1}, \qquad \Delta v_n = -\frac{C}{\omega_0}\left(-1 + \frac{\dot{s}_f \omega_0}{\omega_1 \omega_2}\right) \tag{7.62}$$

$$\Delta y_p = \frac{C}{\omega_2}, \qquad \Delta y_n = -\frac{C}{\omega_0}\left(1 + \frac{\dot{s}_f \omega_0}{\omega_1 \omega_2}\right) \tag{7.63}$$

where,

$$C = \frac{R\dot{s}_f}{2L}, \qquad \omega_1 = \omega_0 + \dot{s}_f, \qquad \omega_2 = \omega_0 - \dot{s}_f, \qquad \omega_0 = \sqrt{\frac{g}{L}}$$

and Δv_p and Δy_p are the changes in v and y due to a positive velocity transition. Δv_n and Δy_n are the changes in v and y due to a negative velocity transition.

As was the case with the ZV2_{Lin} shaper, the shaper parameters are determined by setting the complex response of the two modes equal to zero at the end of the rising and falling portion of the command. The vibration of each mode is found using (7.62) and (7.63) in conjunction the with the free response equations (7.51) and (7.52). Setting the result equal to zero yields:

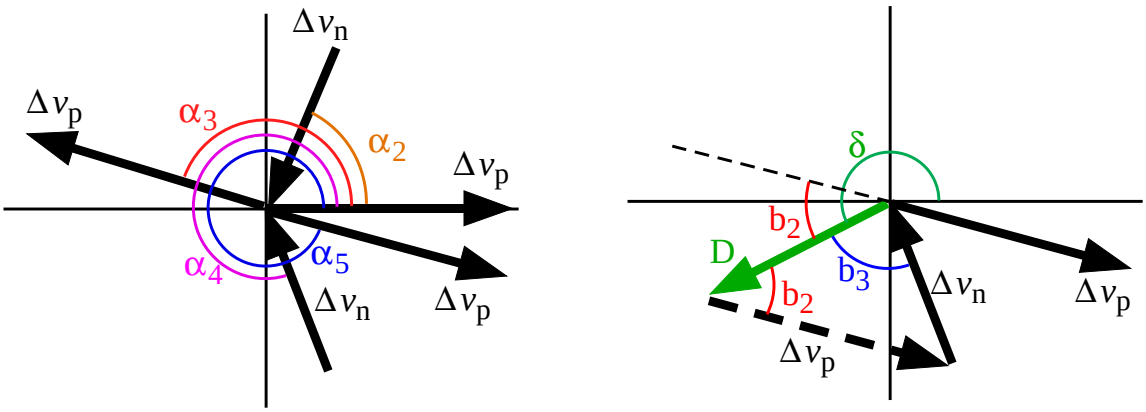
$$\begin{aligned} 0 =& v_p e^{i(\omega_1 t_2 + \omega_n(t_3-t_2) + \omega_1(t_4-t_3)\omega_n(t_5-t_4))} + v_n e^{i(\omega_n(t_3-t_2)+\omega_1(t_4-t_3)\omega_n(t_5-t_4))} + \\ & v_p e^{i(\omega_1(t_4-t_3)\omega_n(t_5-t_4))} + v_n e^{i\omega_n(t_5-t_4)} + v_p \end{aligned} \tag{7.64}$$

$$\begin{aligned} 0 =& y_p e^{i(\omega_2 t_2 + \omega_n(t_3-t_2) + \omega_2(t_4-t_3)\omega_n(t_5-t_4))} + y_n e^{i(\omega_n(t_3-t_2)+\omega_2(t_4-t_3)\omega_n(t_5-t_4))} + \\ & y_p e^{i(\omega_2(t_4-t_3)\omega_n(t_5-t_4))} + y_n e^{i\omega_n(t_5-t_4)} + y_p \end{aligned} \tag{7.65}$$

The shaper times can be solved by simultaneously solving (7.64) and (7.65). Keep in mind that there are really four equations because both the real and imaginary parts of (7.64) and (7.65) must equal zero. Satisfying (7.64) will cause the residual vibration of the first mode to be zero and satisfying (7.65) will cause the residual vibration of the second mode to be zero.

Similar to the ZV2_{Lin} problem, the $\text{UMZV2}_{\text{Lin}}$ problem can be simplified by finding an analytical solution that zeros the vibration of the first mode. The vector diagram representation of the first mode equation, (7.64), is shown in Figure 7.23(a). Notice the alternating pattern of v_p and v_n for the vector lengths. The angle of each vector is α_i, as shown in the figure. Equation (7.64) can be used to relate the α_i angles to the shaper times using a linear transformation:

$$\begin{bmatrix} 0 & 0 & -\omega_n & \omega_n \\ 0 & -\omega_1 & \omega_1-\omega_n & \omega_n \\ -\omega_n & \omega_n-\omega_1 & \omega_1-\omega_n & \omega_n \\ \omega_1-\omega_n & \omega_n-\omega_1 & \omega_1-\omega_n & \omega_n \end{bmatrix} \begin{bmatrix} t_2 \\ t_3 \\ t_4 \\ t_5 \end{bmatrix} = \begin{bmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix} \tag{7.66}$$



(a) Full UMZV2$_{\text{Lin}}$ Vector Diagram

(b) Resolving the Angles

Figure 7.23: Using Vector Diagrams for UMZV2$_{\text{Lin}}$ Commands.

Suppose that angles α_2 and α_3 were already determined. Then, the first three vectors (in quadrants I and II in the figure) can be added together. The resultant vector has length D and angle δ, as shown in Figure 7.23(b). This can be stated mathematically as:

$$De^{i\delta} = \Delta v_p + \Delta v_n e^{i\alpha_2} + \Delta v_p e^{i\alpha_3} \tag{7.67}$$

Now, the goal is to find the angles of the remaining two vectors, α_4 and α_5, so the vectors sum to zero. To make the derivation easier, the Δv_p-vector has been translated so that it touches the ends of the other two vectors, as shown by the dashed vector in Figure 7.23(b). For the vectors to sum to zero, they must form the triangle shown in the third and fourth quadrant. The interior angles of the triangle, b_2 and b_3,can be related to the sides using the law of cosines:

$$(\Delta v_p)^2 = D^2 + (\Delta v_n)^2 - |D||\Delta v_n| cos b_3 \tag{7.68}$$

$$(\Delta v_n)^2 = D^2 + (\Delta v_p)^2 - |D||\Delta v_p| cos b_2 \tag{7.69}$$

Then, the angles b_2 and b_3 can be related back to the vector angles α_4 and α_5:

$$\alpha_4 = \delta + b_3 \qquad\qquad \alpha_5 = \pi + \delta - b_2 \tag{7.70}$$

Given angles α_2 and α_3, the remaining angles α_4 and α_5 can be derived using (7.67)-(7.70). What remains is to choose α_2 and α_3 such that (7.65) is satisfied, thus eliminating vibration in the second mode. Similar to the ZV2$_{\text{Lin}}$ shaper, this was achieved with numerical

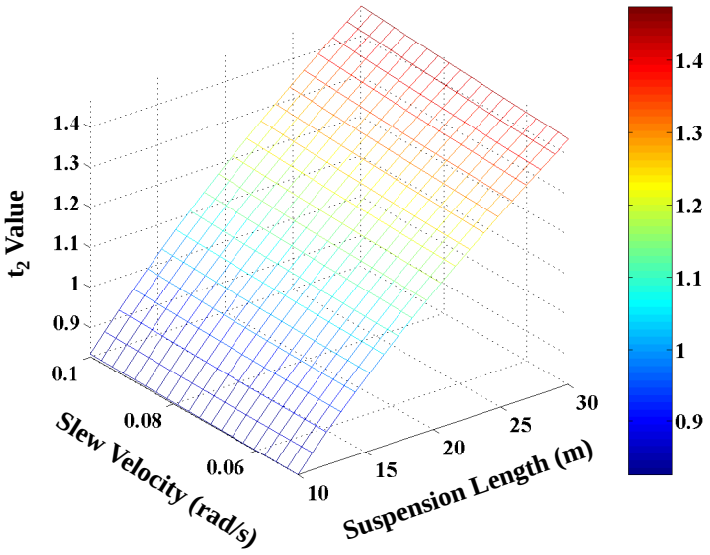


Figure 7.24: t_2 values of UMZV2$_{\text{Lin}}$ Command for Various Parameters.

minimization, using Matlab's *lsqnonlin* function.

Figure 7.24 shows the value of t_2 for various values of $\dot{s}_f$ and L. Similar surfaces were generated for the other command switch times. These surfaces were then fit with a 2^{nd} degree, 2-dimensional polynomial. A 2^{nd} degree polynomial was chosen because it was the lowest degree polynomial that yielded acceptable error (1% or less). The polynomial fit was performed for a system parameter range of: $\dot{s}_f = [0.05 \ldots 0.1]$ (rad/sec) and $L = [10 \ldots 30]$ (m). These ranges were chosen based on the system parameters for industrial tower cranes. The coefficients for this polynomial are given in Appendix D.2. The whole process was then repeated to find the switch times for the falling portion of the UMZV2$_{\text{Lin}}$ command.

A simulation of the UMZV2$_{\text{Lin}}$ command and response appears in Figure 7.25. The simulation parameters were: $L = 20$ m, $\dot{s} = 0.08$ rad/s, $R = 20$ m, and $t_p = 15$ s. Note that these parameters are the same as the ZV2$_{\text{Lin}}$example shown earlier in Figure 7.21. Figure 7.25(a) shows the UMZV2$_{\text{Lin}}$command. Figures 7.25(b) and 7.25(c) show the response of the physical coordinates $\phi, \dot{\phi}, \theta$, and $\dot{\theta}$. Note that the residual vibration is very low during the constant slew velocity segment in the middle of the command, and at the end of the command. The small residual vibration appearing at the end of the command is due to errors

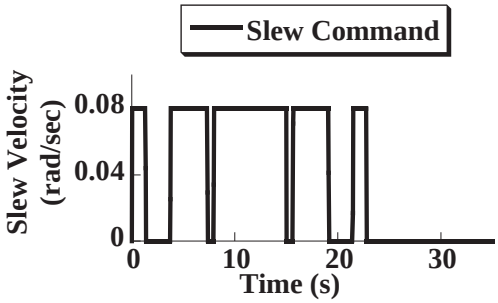
that accumulate over the course of the response. These errors come from deviations between the linearized model and the actual nonlinear model. Figures 7.21(d) and 7.21(e) show the response of the z_1, z_2, z_3, and z_4 coordinates. Note that these values are approximately zero during the constant slew velocity segment in the middle of the command, and at the end of the command. This indicates that the UMZV2$_{\text{Lin}}$ command minimizes the vibration in the two linearized modes.

Compared to the ZV2$_{\text{Lin}}$ response shown earlier in 7.21, the residual vibration of the UMZV2$_{\text{Lin}}$ response is larger. This is because unity-magnitude shapers tend to be more sensitive to errors. The UMZV2$_{\text{Lin}}$ response also forces the system to accelerate faster, thereby causing greater angular deflections as seen in the figures. These larger deflections cause more errors between the linearized model and the nonlinear model which, in turn, creates more residual vibration. However, the UMZV2$_{\text{Lin}}$ command has the advantage of reaching its steady-state velocity, and returning to rest faster than the ZV2$_{\text{Lin}}$ command. For example, the ZV2$_{\text{Lin}}$ command in Figure 7.21(a) has a duration of 25.75 s, whereas the UMZV2$_{\text{Lin}}$ command in Figure 7.25(a) has a duration of 22.75 s.

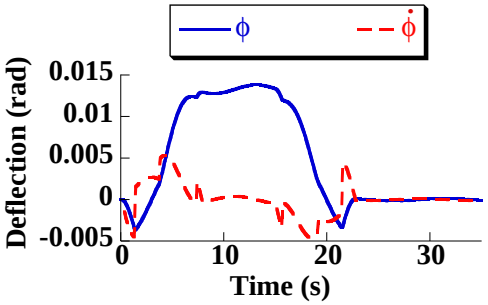
7.3.4.4 Advantages and Disadvantages of Linearized Shapers

The advantages of using linearized shapers are:

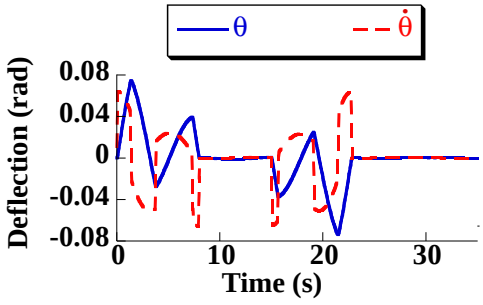
- There is no required radial motion, so the radial position is guaranteed to stay the same.
- The lack of coordinated, multi-axis motion also makes it easier to implement on a real system.
- The linearized shapers provide better vibration suppression than the previous shapers. The standard shapers and radial-assist shapers make assumptions about the tower crane dynamics that are somewhat inaccurate. The assumptions made for linearized shapers are much more accurate.
- This technique can be expanded to form a whole range of new shapers. The ZV2$_{\text{Lin}}$ and UMZV2$_{\text{Lin}}$ are based on 2-mode ZV, and UMZV linear shapers. However, the



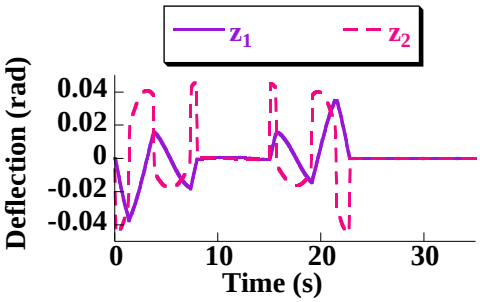
(a) UMZV2$_{\text{Lin}}$ Command



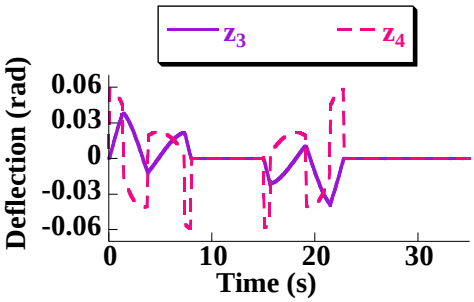
(b) ϕ and $\dot{\phi}$



(c) θ and $\dot{\theta}$



(d) z_1 and z_2



(e) z_3 and z_4

Figure 7.25: UMZV2$_{\text{Lin}}$ command and response.

process for forming the complex response and vector diagrams of the two modes holds for any staircase command.

The disadvantages of linearized shapers are:

- The shapers will not work for short pulse times. More precisely, if the pulse time is less than the duration of the "rising" portion of the command then the command is undefined. Under these circumstances one would switch to using a standard linear shaper.

- Finding the shaper parameters can be computationally expensive. Both the ZV2_{Lin} and $\text{UMZV2}_{\text{Lin}}$ shapers required a numerical optimization process in the final step. In an effort to alleviate this computational burden, the shaper parameters for both shapers were fit to polynomial curves. Using these curves make implementation comparable to well known shapers. However, these curves have a limited range.

7.3.5 Simulation Results

Multiple simulations were run to compare the strengths and weaknesses of the shapers discussed above. There were eight different shapers tested and they can be put into three categories:

Standard Shapers: ZV, UMZV, ZVD, ZV2. These shapers are called "standard" because they come from basic input shaping theory for linear systems. For the tower crane simulations, the ZV, UMZV, and ZVD shapers were designed for a frequency of $\sqrt{g/L}$. The ZV2 shaper was a 2-mode, ZV shaper designed for the linearized frequencies at one-half the final slew velocity, $\dot{s}_f$.

Radial Assist Shapers: $\mathbf{ZV_R}$, $\mathbf{UMZV_R}$. These shapers use combined slewing and radial motion and are modeled after ZV and UMZV shapers respectively.

Linearized Shapers: $\mathbf{ZV2_{Lin}}$. $\mathbf{UMZV2_{Lin}}$. These shapers use only slewing motion and are based on the linearized model. The first shaper is modeled after a 2-mode, ZV shaper, and the second is modeled after a 2-mode, UMZV shaper.

The shapers were compared using three different performance measures:

Residual Vibration A measure of the payload vibration at the completion of a shaped command.

Shaper Duration A measure of the amount of time it takes for the shaped command to reach the commanded velocity.

Robustness A measure of the sensitivity of the shaped command to variations in system parameters.

The results for each performance measure are presented in the three sub-sections below. In addition, a more precise definition of each performance measure is also discussed.

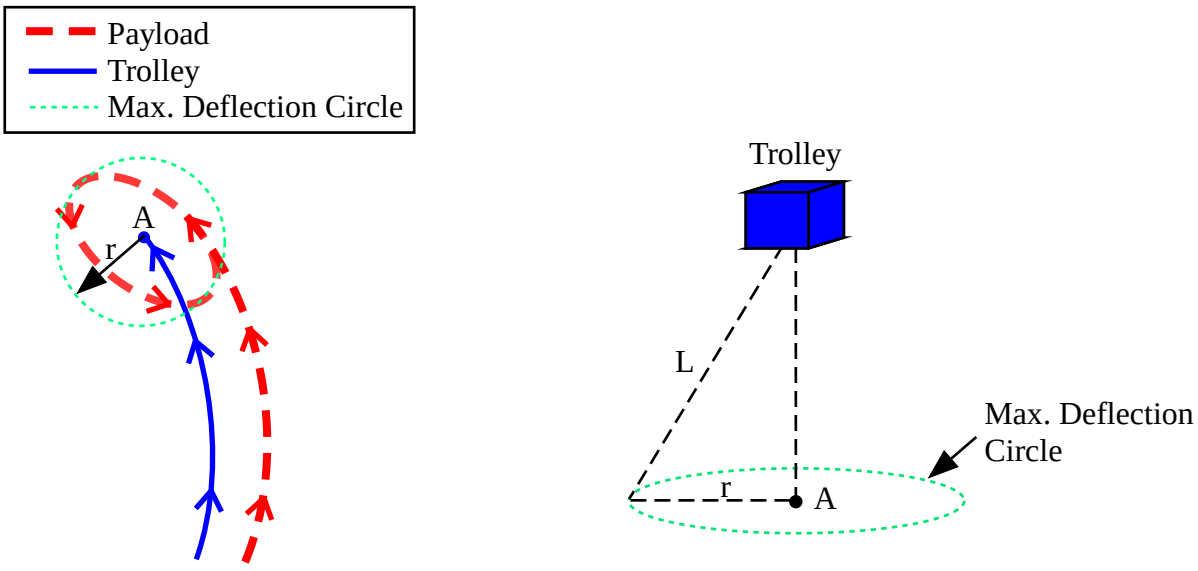
The performance of each shaper is compared across the following independent variables: the final slew velocity, $\dot{s}_f$, the suspension length, L, and in some cases the pulse duration, t_p. Recall that the trolley radial position, R, only has the effect of scaling the vibration. Because this effect is known ahead of time, it will not be changed for the simulations. The desired motion of each simulation is a rest-to-rest slew maneuver with a pulse time, t_p. In every single simulation set, all parameters were kept constant except for the independent variable being studied. Unless otherwise noted, the nominal system parameters used for the each simulation were those given in Table 7.1.

Table 7.1: Nominal System Parameters.

Parameter	*Symbol*	*Value*	*Units*
Suspension Length	L	25	(m)
Final Slew Velocity	$\dot{s}_f$	0.08	$\left(\frac{rad}{s}\right)$
Slew Acceleration	$\ddot{s}_{max}$	5	$\left(\frac{rad}{s^2}\right)$
Initial Trolley Radial Position	R	20	(m)
Max Radial Velocity	$\dot{R}_{max}$	0.25	$\left(\frac{m}{s}\right)$
Radial Acceleration	$\ddot{s}_{max}$	1	$\left(\frac{m}{s^2}\right)$
Pulse Time	t_p	13	(s)

7.3.5.1 Residual Vibration

A series of simulations were run to compare the residual vibration of the seven shapers. The residual vibration of each simulation was defined as the maximum deflection angle at



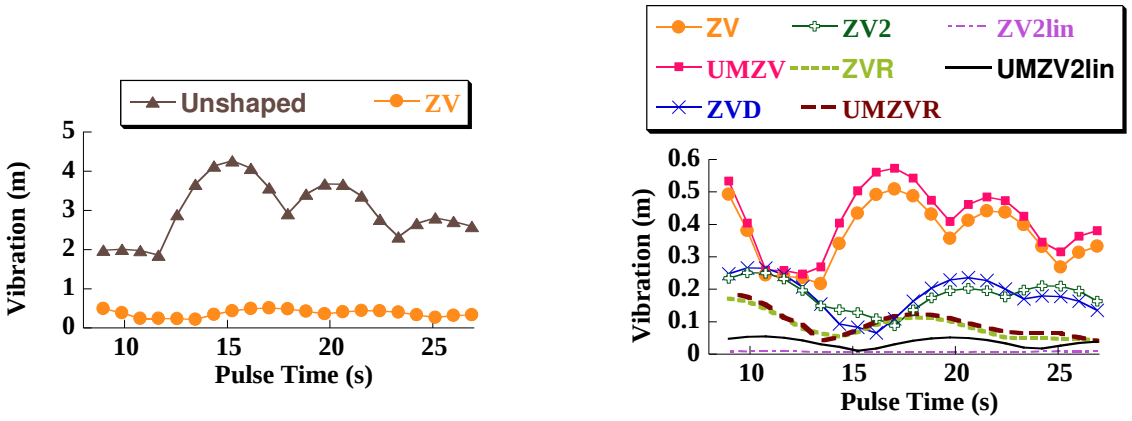
(a) Top View (b) Side View

Figure 7.26: Residual Vibration Measurement

the completion of the motion. Figure 7.26 shows how this angle was calculated. Figure 7.26(a) shows a top view of a typical simulation as the trolley arrives at its final location. The solid line is the trolley position and the dashed line is the payload. At the completion of the maneuver, the payload swings in an elliptical pattern. The major radius of this ellipse, r, defines the maximum deflection circle, shown as the dotted line. Notice that at the completion of the trolley motion, the payload remains within this circle at all times. Figure 7.26(b) shows a side view. L is the suspension length and r is the major radius shown previously. The distance r is the parameter used to measure the residual vibration amplitude in each of the trials described below.

Figure 7.27 shows the residual vibration amplitude of shaped and unshaped commands for various pulse times. All of the other motion parameters were fixed at the values given in Table 7.1 for this set of simulations. On the left, Figure 7.27(a) compares the unshaped and ZV shaped residual vibration. Notice that even with ZV shaping there is a average 87.7% reduction in vibration, a significant reduction in vibration. Figure 7.27(b) shows the residual vibration for all of the shaped commands. The average reduction in vibration of each shaper compared to the unshaped motion is shown in Table 7.2.

The performance of the shapers can be ordered from best to worst as shown in Table



(a) Unshaped and ZV Residual Vibration

(b) Residual Vibration of Various Shapers.

Figure 7.27: Residual Vibration of Shaped and Unshaped Commands for Various Pulse Times.

Table 7.2: Average Vibration Reduction Compared to Unshaped Motion.

% Vibration Reduction	ZV 87.70%	UMZV 86.29%	ZVD 93.97%	ZV2 93.96%
% Vibration Reduction	ZV_R 97.15%	$UMZV_R$ 96.86%	$ZV2_{Lin}$ 99.75%	$UMZV2_{Lin}$ 98.80%

7.3. Shapers that share the same cell, in Table 7.3, have approximately the same performance. Note that the two new types of shapers developed from this thesis, radial assist and linearized, are the top performers of the group.

These results are aligned with the theoretical expectations of each shaper. The linearized model is a good model of the system, so the linearized shapers perform well. The $ZV2_{Lin}$ performs slightly better than the $UMZV2_{Lin}$ shapers. This is because the unity magnitude shaper forces the system to accelerate faster causing the response to deviate from the linearized model, as discussed previously.

The radial assist shapers have the next best performance. These shapers assume that the ϕ and θ modes are uncoupled and use "steady-state relaxation" methods to cancel the

Table 7.3: Comparison of Shaper Vibration Performance

best					worst
$ZV2_{lin}$	$UMZV2_{lin}$	ZV_R $UMZV_R$	ZVD ZV2	ZV UMZV	

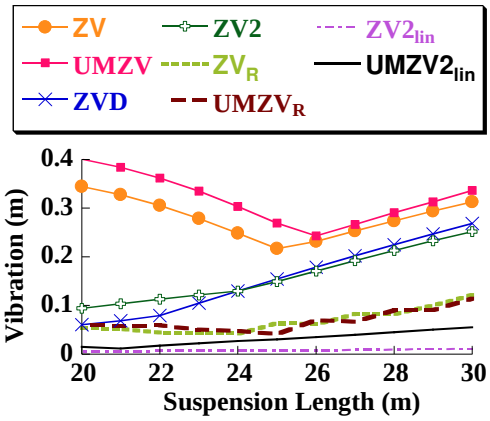
vibration in each direction. This approximation is fairly accurate for small motions which is why these shapers work well. However the linearized model captures the coupling between the two modes, which is why the linearized shapers perform better. In addition, notice that the ZV_R shaper performs slightly better than the $UMZV_R$ shaper. This is caused by the same reasons discussed in the previous paragraph.

The standard shapers have the next best performance. The ZVD and ZV2 shapers is the best performer of the standard shapers due to their robustness properties. However, neither of these two shapers account for the rotational effects of the motion, which is why the linearized and radial assist shapers perform better. The ZV and UMZV shapers are the worst performer due to their poor robustness properties. However these shapers still provide significant vibration reduction compared to the unshaped case. In addition, the UMZV shaper has the added benefit that it is faster than the ZV shaper.

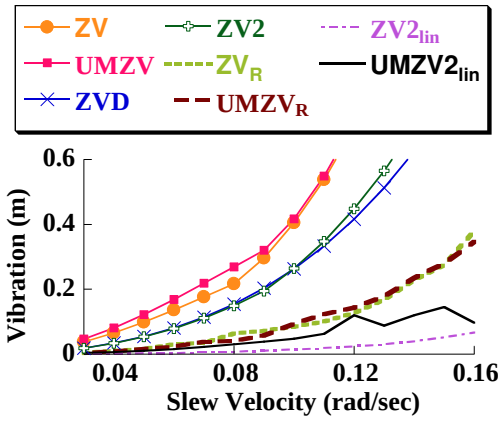
Observe in Figure 7.27(b) that the residual vibration varies in a wave-like pattern as the pulse time changes for all the shapers. The reason is that the residual vibration from the "rising" portion can constructively or destructively interfere with the residual vibration from the "falling" portion of the command depending on the value of the pulse time[1]. However one must be careful about drawing false conclusions from these trends. For example, at $t_p =$ 11 s the residual vibration of the ZV shaper becomes nearly the same as the ZVD shaper, which would seem to contradict the conclusions made above. But a shapers performance should not depend on move distance, so it would be erroneous to say that the ZV and ZVD shapers have equal performance since for most pulse times the ZVD shaper performs better than the ZV shaper. Similar statements could be made about other "minor" trends in the figure. The important trend is the ranked performance of each shaper, which was listed earlier. This destructive and constructive interference causes similar "minor" trends in simulation results presented later in this section.

To test the validity of the above conclusions, the performance of the shapers was also compared by varying other system parameters. Figure 7.28(a) shows the vibration for

[1]Note that this is a simplification. Since the system is non-linear, constructive and destructive interference don't exist in the traditional sense.



(a) Residual Vibration vs. Suspension Length



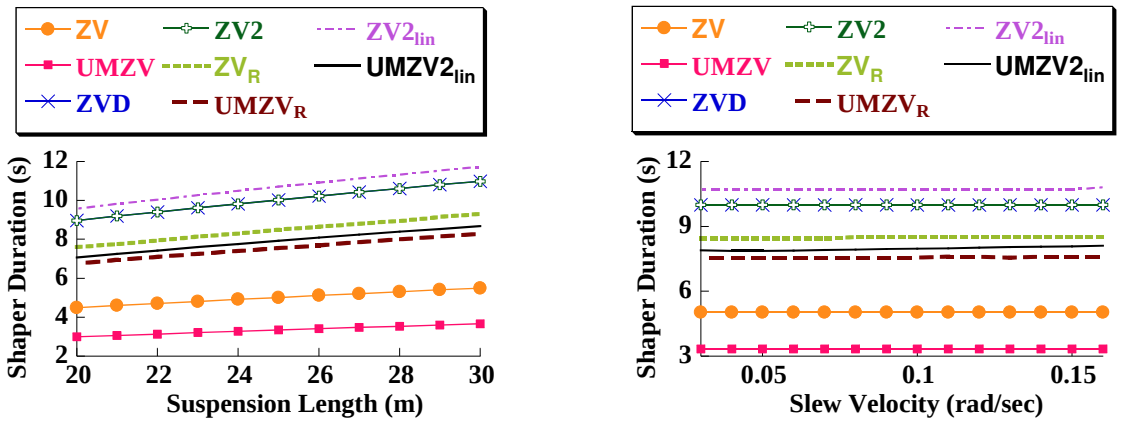
(b) Residual Vibration vs. Slew Velocity

Figure 7.28: Shaper Vibration for Various Parameters

various suspension lengths and Figure 7.28(b) shows the vibration for various slew velocities. Notice that both of these figures support the results in Table 7.3 regarding the relative performance of each shaper. In addition, Figure 7.28(b) shows a interesting trend: as the slew velocity decreases the performance of all the shapers improves. There is a simple explanation for this; moving the system slowly induces less vibration, regardless of the shaper.

7.3.5.2 Shaper Duration

A series of simulations were run to compare the duration of each shaper. The shaper duration is important because a poorly chosen shaper can cause excessively long commands. Since increasing throughput is one of the main reasons for using input shaping in the first place, long shapers are undesirable. In determining command duration, there are two important measurements: the duration of the "rising" portion of the command, and the duration of the "falling" portion. The duration of the "rising" portion is the amount of time it takes for the command to go from rest to the final, constant slewing condition. For a radial assist command, this must be after the radial move is completed as well. Likewise, the duration of the "falling" portion is the amount of time it takes for the command to return to rest. Ideally, the duration of both of these should be small. Since the "rising" and "falling" portions of tower shaped commands are different, the maximum of the two values



(a) Shaper Duration vs. Suspension Length

(b) Shaper Duration vs. Slew Velocity

Figure 7.29: Shaper Duration for Various Parameters

is reported in this set of tests.

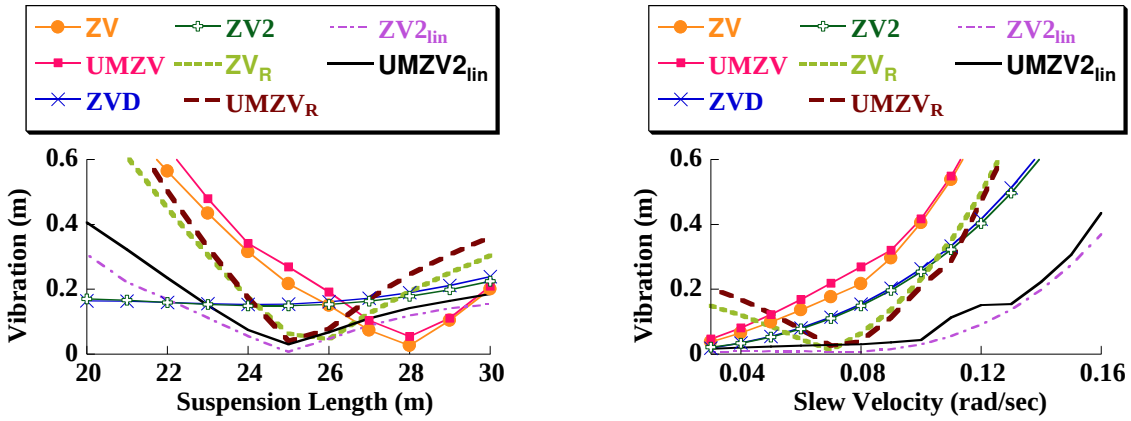
Figure 7.29 shows the duration of all the shapers for various parameters. The left figure, 7.29(a), shows the vibration for various suspension lengths and the right figure, 7.29(b), shows the vibration for various slew velocities. The performance of the shapers, with respect to duration, can be ordered from best to worst as shown in Table 7.4. Both

Table 7.4: Comparison of Shaper Duration

short							long
UMZV	ZV	$UMZV_R$	$UMZV2_{lin}$	ZV_R	ZV2	ZVD	$ZV2_{lin}$

figures support these same conclusions.

The UMZV and ZV commands are the fastest, with the UMZV being the fastest of the two. However, recall that both these commands induced the most amount of residual vibration of the shapers tested. The two unity magnitude commands, $UMZV_R$ and $UMZV2_{Lin}$, are faster than their ZV counterparts, ZV_R and $ZV2_{Lin}$. Unity magnitude commands are faster because they immediately command the system to move at full speed. This is true for standard, linear shapers, and is also true for these non-linear shapers. The ZV_R command is the next shortest, followed by the ZVD and ZV2 commands. The longest command is the $ZV2_{Lin}$. However, recall that this shaper also had the least amount of residual vibration of all the shapers tested. Note that there is a clear tradeoff between speed and vibration



(a) Sensitivity to Suspension Length

(b) Sensitivity to Slew Velocity

Figure 7.30: Shaper Robustness

suppression for these shapers. This is commonly encountered when designing shapers for both linear and nonlinear systems.

7.3.5.3 Robustness

The robustness of each of the seven shapers was compared. Robustness is how well the shaper performs due to unforseen parameter variation. In these tests each shaper was designed for the same nominal set of system parameters, given in Table 7.1. Then one of the parameters is changed, while the shaper is kept the same. The residual vibration is recorded for each value using the same maximum radial deflection, r, that was used earlier in Figure 7.26. If the shaper is able to retain a low amount of residual vibration than it has a "low sensitivity" to parameter variation and is robust. Robust shapers are desirable because in many cases, particularly for cranes, exact system parameters may not be known. It can also mean that the same shaper can be used even though a parameter changes, saving the cost of installing extra sensors to measure that parameter.

Figure 7.30 shows the sensitivity plots for the shapers. On the left, Figure 7.30(a) shows the sensitivity to variations in the suspension length. On the right, Figure 7.30(b) shows the sensitivity to variations in the final slew velocity, $\dot{s}_f$. The robustness of the shapers depends on what parameter is being changed. With respect to changes in suspension length, the robustness of the shapers are ordered as shown in Table 7.5. The table was formed for

Table 7.5: Comparison of Shaper Robustness to Suspension Length

more robust			less robust
ZVD ZV2	$ZV2_{lin}$ $UMZV2_{lin}$	ZV_R $UMZV_R$	ZV UMZV

sensitivity to large changes in suspension length. However, Figure 7.30(a) shows that for small changes both the linearized and radial assist commands have superior robustness. Deciding which shaper to use for a given application will depend on how much variation in suspension length is expected.

For changes in slew velocity, the robustness is ordered as shown in Table 7.6.

Table 7.6: Comparison of Shaper Robustness to Slew Velocity

more robust			less robust
$ZV2_{lin}$ $UMZV2_{lin}$	ZVD ZV2	ZV_R $UMZV_R$	UMZV ZV

For variations in slew velocity and suspension length, both ZV and UMZV shapers have the least amount of robustness. Notice in Figure 7.30(a) that the ZV and UMZV residual vibration reaches a minimum at $L = 28$. However, this minimum should not be used to evaluate the performance of these shapers since it does not occur for different pulse durations. The minimum occurs because of the destructive interference phenomenon mentioned earlier. As mentioned earlier, the ZVD and ZV2 commands have good robustness for large parameter changes. However, for small parameter changes both the radial and linearized shapers perform better.

7.3.5.4 Short Commands

As mentioned earlier the ZV_R, $UMZV_R$, $ZV2_{Lin}$, and $UMZV2_{Lin}$, commands are all undefined if the pulse time, t_p, is sufficiently short. An alternative is to use a standard linear shaper, ZV, UMZV, ZV2, or ZVD, for such short commands. Figure 7.31 shows the residual vibration of these standard shapers for several short pulse times. The ZVD and ZV2 shaper have less residual vibration than the ZV and UMZV. However, the ZVD and ZV2 also have a longer duration. Choosing which shaper to use depends on the relative importance of shaper duration and vibration suppression for the particular application. Note that as the

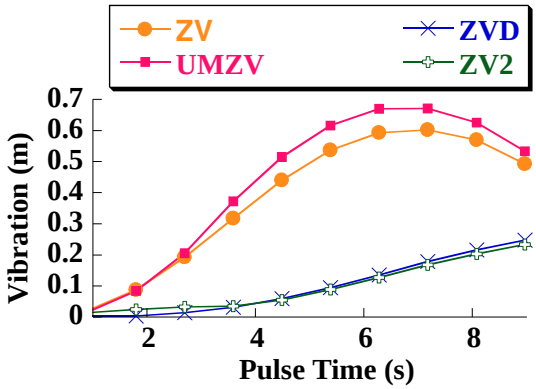


Figure 7.31: Residual Vibration of Standard Shapers for Short Commands

pulse time decreases the performance of these shapers improves. That is because for a short slewing distance, the motion of the system becomes more "linear" and less "rotational". In a real tower crane the system could switch to these shapers if a short move distance was known ahead of time.

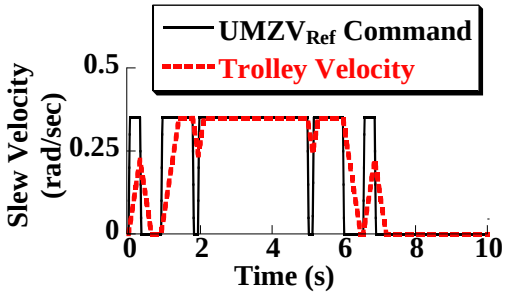
7.3.6 Tower Crane Shaping: Experimental Results

The shaping strategies mentioned above were tested on the tower crane to evaluate their performance on a real system. The nominal system parameters were:

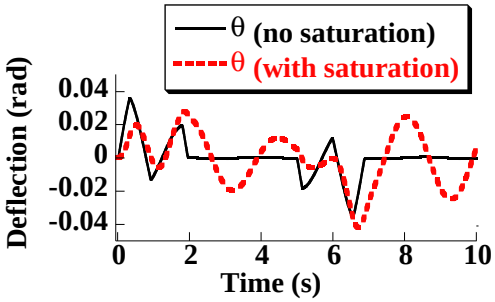
Table 7.7: Nominal Tower Crane System Parameters.

Parameter	*Symbol*	*Value*	*Units*
Suspension Length	L	1	(m)
Final, Setpoint Slew Velocity	$\dot{s}_f$	0.35	$\left(\frac{rad}{s}\right)$
Slew Acceleration	$\ddot{s}_{max}$	0.7	$\left(\frac{rad}{s^2}\right)$
Initial Trolley Radial Position	R	0.9	(m)
Max Radial Velocity	$\dot{R}_{max}$	0.14	$\left(\frac{m}{s}\right)$
Radial Acceleration	$\ddot{s}_{max}$	1.2	$\left(\frac{m}{s^2}\right)$
Pulse Time	t_p	3	(s)

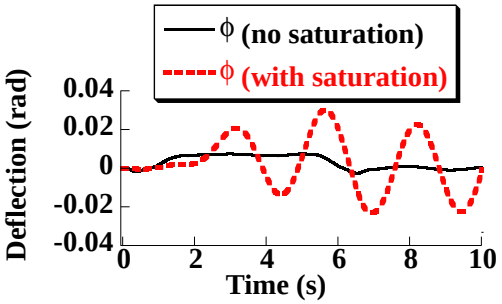
Rather then test all seven shaping strategies, only the ZV, ZVD, ZV_R, and ZV2_Lin commands were tested. The ZV2 command was not tested because the simulations showed that it had comparable performance to the ZVD shaper. The unity magnitude commands, UMZV, UMZV_R and UMZV2_Lin, were not tested due to the acceleration limitations of the tower crane. To understand why, Figure 7.32 shows a simulation of the tower crane with



(a) Desired and Actual Trolley Velocity



(b) θ Response With and Without Acceleration Saturation.

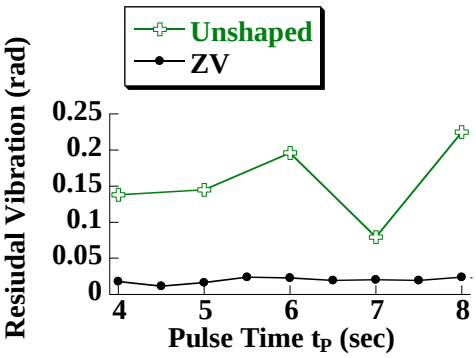


(c) ϕ Response With and Without Acceleration Saturation.

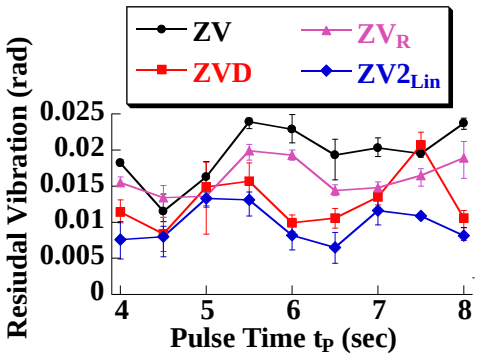
Figure 7.32: Comparison of UMZV2$_{\text{Lin}}$ Command and Response, With and Without Tower Crane Acceleration Saturation.

and without acceleration saturation using a UMZV2$_{\text{Lin}}$command. The simulation uses the actual parameters from the tower crane. Figure 7.32(a) shows the UMZV2$_{\text{Lin}}$ command and the actual trolley velocity. Due to the acceleration limits of the crane, the trolley has a difficult time tracking the rapidly changing UMZV2$_{\text{Lin}}$ command. Figures 7.32(b) and 7.32(c) show the ϕ and θ response to the original UMZV2$_{\text{Lin}}$ command without saturation, and the actual trolley velocity with the saturation. These figures make it clear that the acceleration saturation degrades the shaper performance and makes it unusable for this system. The same argument applies to the UMZV$_{\text{R}}$command.

The experimental results are shown in Figure 7.33. The experiments were performed by moving the crane a set pulse time t_p and measuring the residual vibration using the same angular measure discussed in the previous section. Figure 7.33(a) compares the residual vibration of unshaped and ZV commands. The ZV command reduces the residual vibration



(a) Unshaped verses ZV Commands.



(b) Comparison of Various Shaped Commands.

Figure 7.33: Residual Vibration of Various Commands for Different Pulse Times.

by nearly 88%, a significant reduction. The same reduction in vibration was noted in the simulation results presented earlier.

Figure 7.33(b) shows the residual vibration of the four shaped commands. In this set of experiments, each trial was repeated three times for the same pulse duration. The average was plotted, and the bars show the maximum and minimum values. Table 7.8 shows the percent reduction in vibration of the four shapers averaged across all the pulse times. These values were calculated by comparing the average vibration of each shaper to the average vibration of the unshaped data.

Table 7.8: Experimental Average Vibration Reduction Compared to Unshaped Motion.

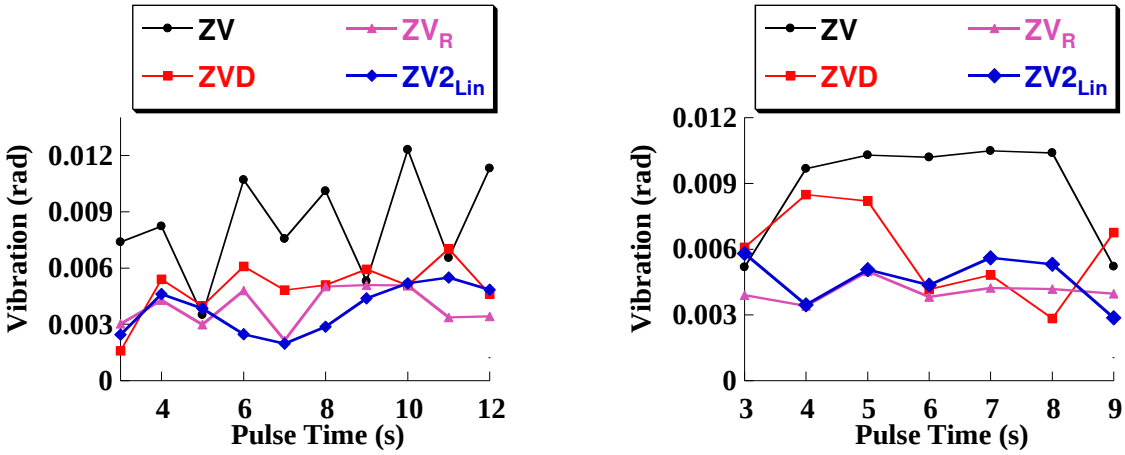
% Vibration Reduction	ZV	ZVD	ZV_R	$ZV2_{Lin}$
	88%	92%	90%	94%

The data presented in Figure 7.33(b) and Table 7.8 supports many of the conclusions from the simulations. The percent reduction in vibration of these shapers, 88%-94%, is comparable to the percent reduction range for the simulated data, 88%-99%. However, the upper bound of this range is lower for the experimental data due to unmodeled experimental dynamics (eg. structural vibration, imperfect trajectory tracking, etc...). The $ZV2_{Lin}$ shaper has the least residual vibration, similar to the simulated data. The ZV shaper is the worst of the shaped commands, although still far better than unshaped, similar to the simulated data. However, the experimental data differs from the simulation results regarding the

performance of the ZVD and ZV_R shapers. The simulation results showed that the ZV_R shaper has lower residual vibration than the ZVD shaper, but the experimental data shows the opposite. This is because the ZV_R shaper uses a more aggressive command than the ZVD shaper which induces structural vibration in the experimental setup. In addition, the small scale of the tower crane limits the nonlinear effects of the rotational motion which makes the ZVD shaper perform better. As the size of the crane increases, particularly the trolley radius, the nonlinear effects of the rotation become more important causing the ZV_Rshaper to outperform the ZVD shaper as shown in the simulations earlier.

It should be noted that all of these experiments were performed remotely using the tower crane in Japan. The feedback, swing-reducer was used to zero the payload swing before every trial. However, this feedback control never completely eliminated the vibration. On average there was between 0.004 rad and 0.005 rad of vibration at the beginning of each trial. This explains why all of the data points are displaced by approximately 0.005 rad. In addition, this initial vibration would sometimes add constructively or destructively to the response. So, repeating a single trial never resulted in exactly the same residual vibration. This explains why the error bars have a maximum value of about 0.005 rad.

The experiments were repeated for different crane configurations. Figure 7.34(a) shows the results for a trolley radius of 0.55 m. Figure 7.34(b) shows the results for a trolley radius of 0.55 m and a suspension length of 1.5 m. In addition, the pulse time range was extended to get a larger range of data. Note that only one trial was performed at for each pulse time. The data in these plots still shows that the ZVD, ZV_R, and $ZV2_{Lin}$ commands are better than the ZV command. However the $ZV2_{Lin}$ command is only marginally better than the others. Part of the reason is that the trolley radius was reduced from 0.9 m to 0.55 m. Recall that the residual vibration scales with the radius. So, at a shorter radius the residual vibration of all the shapers is less making it harder to see an improvement between them.



(a) L=1 m, R=0.55 m

(b) L=1.5 m, R=0.55 m

Figure 7.34: Experimental Data: Residual Vibration of Various Shaped Commands for Different Tower Crane Configurations.

7.4 Tower Crane Study Conclusions

This chapter addressed input shaper for tower cranes. The equations of motion for a tower crane were derived and linearization revealed that the system has two modes of vibration for non-zero slew velocity. The steady state configuration of the system was also derived.

Two new shaping strategies were developed. The first strategy is based on "steady-state relaxation" and assumes the radial and tangential vibration are uncoupled. The system is moved using a standard linear shaper, and then trolley moves radially to bring the system into steady state. Two commands were developed using this strategy: a ZV_R command and a on-off $UMZV_R$ command.

The second shaping strategy is based on the linearized model. A vector-based method for predicting the residual vibration of a rapidly-changing staircase command was developed. Shaped commands were then formed by choosing the amplitudes and times such that the residual vibration was eliminated. Two commands were developed using this strategy: a $ZV2_{Lin}$ and a on-off $UMZV2_{Lin}$ command.

The newly developed commands were compared to standard shapers both in simulation and experiments. The commands were compared using the following performance measures: residual vibration, command duration, robustness, and computational complexity. Table

Table 7.9: Performance of Various Shapers.

	best							worst
Vibration	$ZV2_{lin}$	$UMZV2_{lin}$	ZV_R	$UMZV_R$	ZVD	ZV2	ZV	UMZV
Duration	UMZV	ZV	$UMZV_R$	$UMZV2_{lin}$	ZV_R	ZV2	ZVD	$ZV2_{lin}$
Robustness to Length	ZVD	ZV2	$ZV2_{lin}$	$UMZV2_{lin}$	ZV_R	$UMZV_R$	ZV	UMZV
Robustness to Slew Velocity	$ZV2_{lin}$	$UMZV2_{lin}$	ZVD	ZV2	ZV_R	$UMZV_R$	UMZV	ZV
Computation	ZV	ZVD	ZV2	UMZV	ZV_R	$UMZV_R$	$ZV2_{lin}$	$UMZV2_{lin}$

7.9 summarizes the results. The shapers are ordered from best to worst. Two or more shapers in the same rectangle indicates comparable performance. The table is taken from the simulation results using the industrial sized crane model. Most of the experimental results validate these conclusions. However, there were some minor discrepancies between the simulated and experimental results due to the small scale of the setup, and other experimental effects explained earlier.

The ZV_R, $UMZV_R$, $ZV2_{Lin}$, $UMZV2_{Lin}$ commands rank highly in most of the categories. The $ZV2_{Lin}$ command has the least amount of vibration, while the $UMZV_R$ has the shortest duration (not including the ZV command, since its performance is so poor). However, standard shapers require much less computation to derive than the newly developed shapers. It should also be noted that all of these shapers provided at least a 88% reduction in residual vibration compared to the unshaped case.

The experimental results also demonstrated that cranes with slow slew accelerations may be incapable of using $UMZV_R$ or $UMZV2_{Lin}$ commands. However, many crane systems may be able to use these commands. The experiments were performed on a small-scale crane with a relatively short period. On a industrial-size crane the period would be much longer, creating $UMZV_R$ and $UMZV2_{Lin}$ commands that are easier for the system to track.

In summary, given the requirements of any tower crane application, the results in this chapter can be used to choose the best shaper. Table 7.9 can aid in this process. For example, if speed is most important and only a moderate amount of vibration suppression is required, a $UMZV_R$shaped command might be the best solution. However, if the structure

is incapable of the rapid acceleration, and the user desires a pure slewing motion that is easy to program, then a ZV shaped command might be best. On the other hand, if position accuracy is extremely important and shaper duration is unimportant then the $ZV2_{Lin}$command might be the best choice.

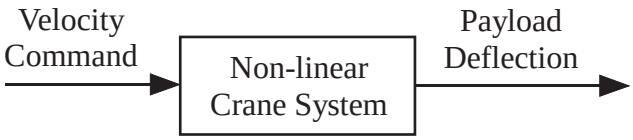
CHAPTER VIII

INPUT SHAPING FOR DRIVE NONLINEARITIES

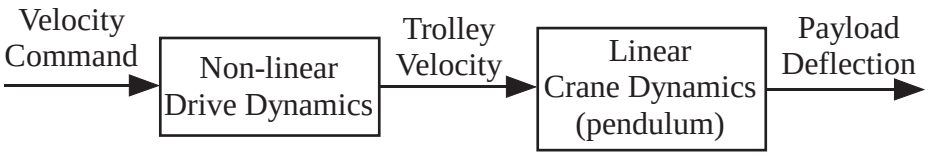
The vector diagram approach was originally pioneered by Seering and Singhose as a simple means for solving for input shapers [80,84,85]. Rather than solve a complex set of algebraic equations, deriving input shapers is transformed into a geometric problem which usually has a simpler solution method. This chapter strives to achieve the same goal, but for nonlinear systems instead.

Until now, the vector approach has only been limited to linear systems. However, the vibration of certain nonlinear systems can also be represented as vectors, thereby lending itself to a similar geometric approach. Furthermore, because the vector geometry of linear input shaping has already been solved to a large extent, the nonlinear solutions can build on these pre-determined solutions.

One idea that is critical to the development of this chapter is how a nonlinearity manifests itself in a system. Consider, for example, a nonlinear crane system as shown in Figure 8.1(a). The goal is to choose a command that makes the crane move with as little residual vibration as possible. In this form, the problem is extremely difficult because so little is known about the nonlinear block.



(a) Full Nonlinear Crane.



(b) Partially Nonlinear Crane.

Figure 8.1: Different Representations of Nonlinear Cranes

Now consider the nonlinear crane system shown in Figure 8.1(b). All of the nonlinear elements are represented in the "nonlinear drive dynamics" block. This might represent friction, the dynamics of the power circuitry, the effects of the drive controller, or several other nonlinear elements. The other block represents pendulum dynamics of the crane, which can be well approximated as linear. This block diagram lends itself to many more solutions than the first one. Because the pendulum dynamics of the crane are linear, standard input shaping concepts can be used to determine trolley velocity profiles that yield zero vibration or good robustness. This information can then be used to solve for velocity commands by compensating for the nonlinear drive dynamics. That is precisely the strategy that will be developed in this chapter.

The algorithms developed in this chapter make several assumptions about the nature of the nonlinear block shown in Figure 8.1(b):

1. The nonlinearity is time invariant.

2. The nonlinearity is input-output bounded.

3. The input to the nonlinear element is always a staircase command. In other words, the command consists of instantaneous velocity transitions and constant velocity segments.

4. When given a staircase command the nonlinear response settles in a finite length of time. Note that the settling time can change depending on the command, but it must be finite.

These assumptions are consistent with the properties of real-crane systems. Most drive systems are time-invariant and input-output bounded, consistent with assumptions #1 and #2. Assumption #3 arises from the fact that most crane systems are driven with toggle buttons that generate pulse signals. This pulse signal can be considered a one-step staircase command. Furthermore, all of the velocity commands in this chapter are staircase commands. Assumption #4 states that the trolley velocity reaches a constant velocity in a finite amount of time, which is true for properly designed crane systems.

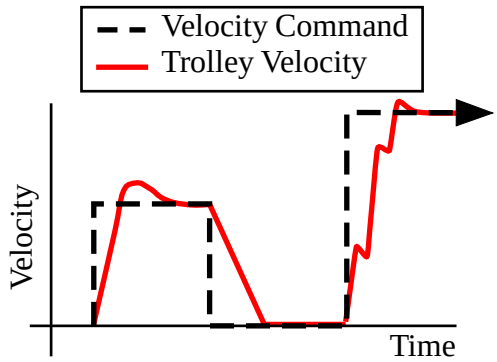


Figure 8.2: Example of a Velocity Command and the Nonlinear Trolley Response.

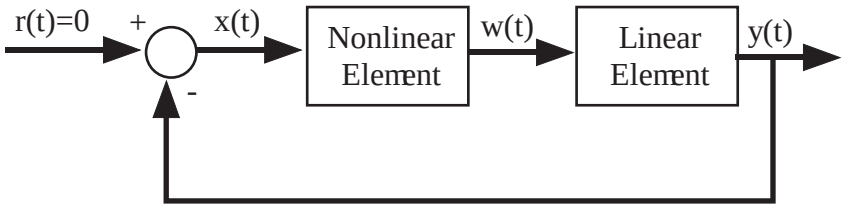
Figure 8.2 shows an example of what the input and output of this nonlinear block could look like. The solid line is the commanded velocity and the dashed line is the actual trolley velocity, v_t. Recall that these signals are the input and output of the nonlinear block in Figure 8.1(b). Note that the input and output are consistent with the assumptions above.

The techniques developed in this chapter are similar to another common nonlinear analysis tool, describing function analysis. However, there are some subtle differences that are worth mentioning. Figure 8.3(a) shows the block diagram used for describing function analysis [91]. Below it, Figure 8.3(b) shows the block diagram used for the nonlinear drive analysis in this chapter. In describing function analysis the system is assumed to oscillate at a single frequency and the effect of the nonlinear element is described using a amplitude and phase shift. The main application of this analysis is for predicting limit cycles and more details are given in [91].

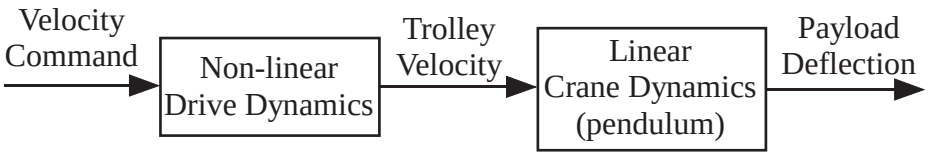
The analysis techniques developed in this chapter are similar to describing function analysis in three ways:

- The system is separated into a linear block and a nonlinear block.
- The system response is assumed to be sinusoidal, oscillating at a single frequency.
- The effect of the nonlinearity is represented as a amplitude and phase shift.

However, there are several differences between the techniques developed in this chapter and describing function analysis:



(a) Block Diagram Used for Describing Function Analysis.



(b) Block Diagram Used for Nonlinear Drive Analysis in this Chapter.

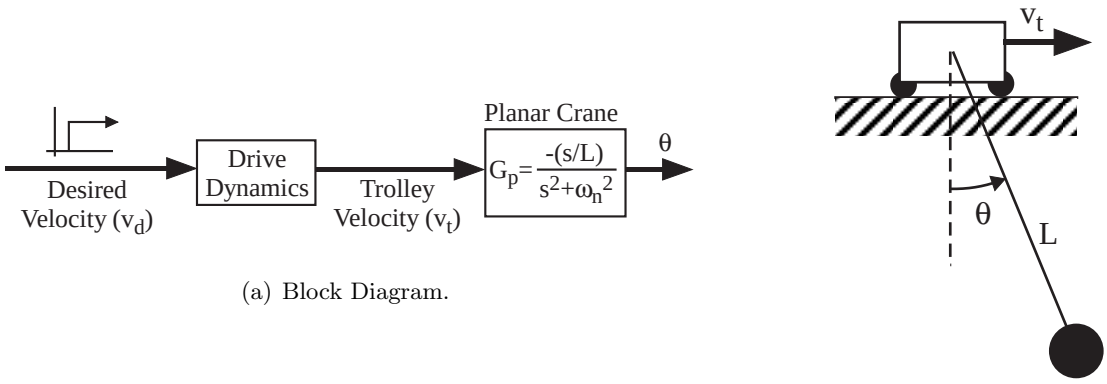
Figure 8.3: Describing Function Analysis verses Nonlinear Drive Analysis.

- Describing function analysis assumes that both the linear and nonlinear blocks are contained within the feedback loop. However, in the nonlinear drive model the feedback loop is assumed to be contained within the nonlinear drive block and the linear block is outside of the feedback loop.

- Describing function analysis usually assumes that the reference command, r(t), is zero. In contrast, the input to the nonlinear drive model is a staircase command.

- Describing function analysis examines the steady-state response of the nonlinear element to a sinusoidal command. This is a direct consequence of the feedback signal in Figure 8.3(a). However, in the nonlinear drive analysis this feedback is not present. Instead, the steady-state response of the linear system to the nonlinear command is examined. This steady-state response is assumed to be sinusoidal because the linear system is a planar crane.

The rest of this chapter discusses the theory behind the vector-based solutions and shows examples of it being applied to various systems. First, vector-based solutions are developed for continuous systems. Second, this strategy is applied to cranes with a braking nonlinearity and some experimental and simulation results are presented. Third, the vector-based approach will be extended to digital systems. Fourth, the digital approach will be applied to a simulated nonlinear system as well as an experimental crane setup.



(a) Block Diagram.

(b) Planar Crane.

Figure 8.4: Block Diagram and Plant for Single Step Simulation.

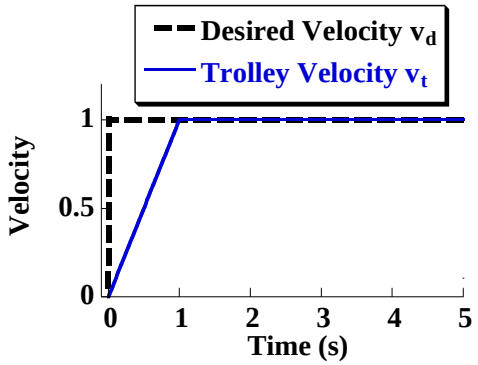
8.1 Vector Approach for Continuous Systems

The main focus of this chapter is to use vectors to design commands for crane systems with nonlinear drives. The approach is targeted at crane systems, but can easily be expanded to other systems. First, the response to a single step command will be discussed. Then, this will be used to form a systematic way of analyzing shaped commands. A new, analytical method for deriving nonlinear shapers is developed by matching this response to a standard shaped response. The limitations and physical meaning of the process are also discussed.

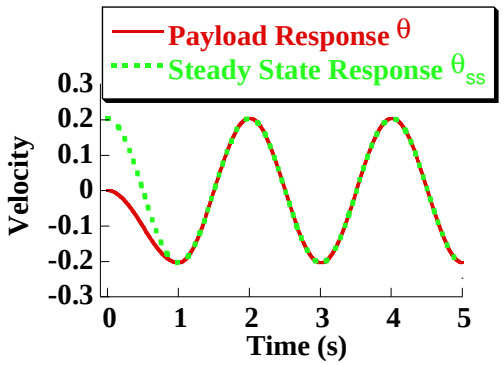
8.1.1 Single Step Response

Figure 8.4(a) shows a block diagram for a simple crane system. The desired velocity command, v_d, is a step. This command is sent to the drive system. The drive consists of the control electronics, power electronics, and mechanical components that move the trolley. The drive system cannot track the desired velocity exactly, so the actual trolley velocity is described by a separate signal, v_t. This signal drives the plant, which is a planar crane model. The planar crane is shown in Figure 8.4(b) with a suspension length L. The input is the trolley velocity, v_t, and the output is the deflection angle, θ. The transfer function for a planar crane, as shown in Figure 8.4(a), is:

$$G_p(s) = \frac{\theta(s)}{v_t(s)} = -\frac{\frac{1}{L}s}{s^2 + w_n^2} \tag{8.1}$$



(a) Trolley Velocity.



(b) Response.

Figure 8.5: Single Step Command and Response.

where $w_n = \sqrt{g/L}$ is the natural frequency of the crane. It is assumed that there is zero damping in the payload response and the cable length is fixed. It is also assumed that the payload deflection angle is small.

Suppose the crane drive system contained an acceleration limit. Figure 8.5(a) shows the desired and actual trolley velocity commands for this case. Because the trolley velocity is acceleration limited, the final desired velocity of 1 m/s is not attained until t=1 second. The resulting payload response is shown in Figure 8.5(b). The steady-state portion of the response begins at $t = 1$ second, when the trolley velocity reaches its steady-state value. The dotted line shows the steady-state response, $\theta_{ss}(t)$. Because most input shapers reduce or eliminate the vibration during the steady-state portion of the response, the focus of this section is being able to predict and control $\theta_{ss}(t)$.

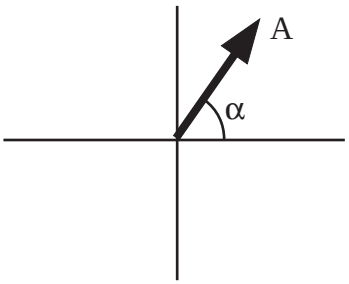
The first step is to determine the amplitude and phase of $\theta_{ss}(t)$. Suppose the Laplace transform of the trolley velocity exists, and is given by $v_t(s)$. It follows that the payload response is given by:

$$\theta(s) = G_p(s)v_t(s) = -\frac{\frac{1}{L}s}{s^2 + w_n^2}v_t(s) \tag{8.2}$$

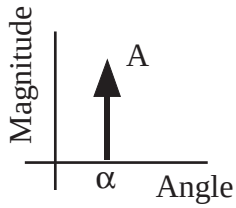
Rearranging the terms:

$$\theta(s) = \underbrace{-\frac{sv_t(s)}{L\omega_n}}_{C(s)} \underbrace{\frac{\omega_n}{s^2 + \omega_n^2}}_{\text{Sine Input}} \tag{8.3}$$

Note that the second term is the Laplace transform of a sine wave with natural frequency ω_n. In this form the "input" is a sine wave with frequency, ω_n, and the "system" is C(s).



(a) Polar Plot.



(b) Angle-Magnitude Plot.

Figure 8.6: Plotting the Vibration Vectors.

The steady state response can be simply derived by setting $s = j\omega_n$ and evaluating the phase and magnitude of C(s) [64]. Therefore, the steady-state payload response is given as:

$$\theta_{ss}(t) = \frac{|v_t(j\omega_n)|}{L} sin\Big(\omega_n t - \frac{\pi}{2} + \angle v_t(j\omega_n)\Big) \tag{8.4}$$

Where $v_t(j\omega_n)$ is the complex number resulting from setting $s = j\omega_n$ in $v_t(s)$ and $|v_t(j\omega_n)|$ and $\angle v_t(j\omega_n)$ represent the magnitude and angle of this complex number.

The steady-state response can be written using more compact notation. The important parameters for describing the steady-state response are its magnitude and phase. This can be captured as a vector or phasor:

$$\vec{v} \equiv A\angle(\alpha) \tag{8.5}$$

where A is the magnitude of the steady-state response and α is the phase of the steady-state response. Note that the frequency of the response is not reported since it is assumed to be the crane's natural frequency ω_n. Applying this notation to (8.4) yields:

$$\vec{v} = \frac{|v_t(j\omega_n)|}{L} \angle \Big(-\frac{\pi}{2} + \angle v_t(j\omega_n)\Big) \tag{8.6}$$

This vector notation also offers a graphic means of measuring the vibration. For example, the vector can be drawn on a polar plot, as shown in 8.6(a). In this figure, the magnitude of the vector is A and the angle is given by α. The magnitude and angle can also be plotted in cartesian coordinates, as shown in Figure 8.6(b). The graphical representations will be discussed more in later sections.

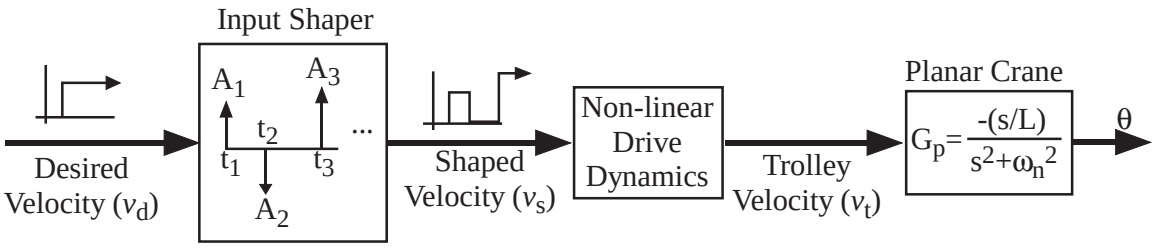


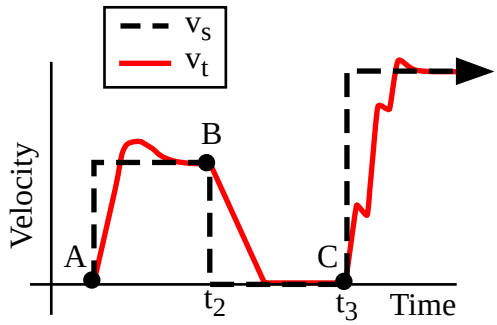
Figure 8.7: Block Diagram for input-shaped Crane

8.1.2 Multi-Step Response

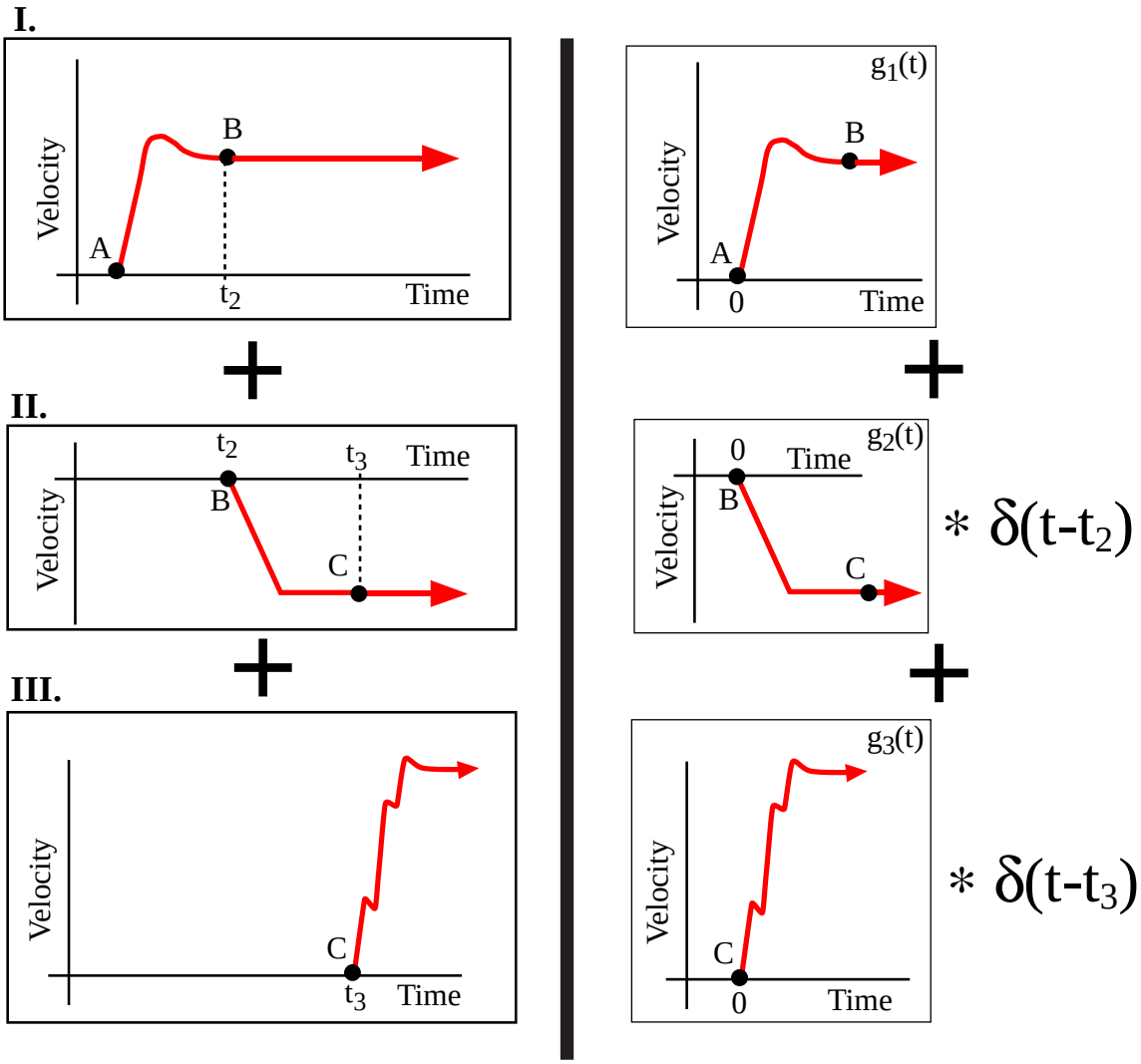
When an input shaper is applied to a step command, a multi-step command results. This section will examine the crane response to a multi-step input by building on the previous concepts. Figure 8.7 shows the block diagram for a crane system with input shaping. The desired velocity, v_d, is a step. An input shaper modifies this command and it becomes the shaped velocity command, v_s. The input shaper uses impulse amplitudes, $[A_1, A_2, \ldots]$, and impulse times, $[t_1, t_2, \ldots]$. The shaped velocity command is sent to the drive system, resulting in the actual trolley velocity, v_t. This trolley velocity drives a planar crane using the same transfer function as shown earlier. The output is the payload deflection θ.

The drive dynamics modify the shaped velocity command. Figure 8.8(a) shows an example of this effect. The dashed line is the shaped velocity, v_s, and the solid line is the trolley velocity, v_t. Notice that the relationship between the two commands is clearly nonlinear, due to the assumed nonlinear dynamics of the drive system.

A systematic method for analyzing the steady-state response to this command can be developed. The method begins by dividing the command into segments. In the figure these segments are A-B, B-C, and C-∞. The last segment starts at C and goes on for infinity. A new segment begins when the shaped command transitions to a new step. In other words, each letter coincides with a shaper impulse time. Note that the example in the figure shows that $v_t = v_s$ at these points, but this need not be the case. These segments are used to break the total command into pieces. In this example the command is broken into three pieces labeled I-III in Figure 8.8(b). Piece I is formed by taking segment A-B and extending the value at B for an infinite length of time. Piece II is formed by taking segment B-C, offsetting



(a) Original Command



(b) Command Breakdown 1.

(c) Command Breakdown 2.

Figure 8.8: Breaking Down Command for Analysis.

all the values so the segment begins with zero magnitude, and extending the offset value at C for an infinite length of time. Piece III is formed by taking segment C-∞, and offsetting all the values so the segment begins with zero magnitude. The process could be easily adapted for any n-step, staircase velocity command. The sum of these pieces reconstruct the original trolley velocity profile, v_t.

The next step is to shift the time of each piece so it starts at time $t = 0$. Figure 8.8(c) shows this step. Piece I already starts at zero so there are no modifications made. Piece II is time-shifted and then convolved with a delta-dirac function delayed by t_2. Note that the result of these convolved terms is the original segment II shown in 8.8(b). The same procedure is carried out for piece III. These newly, formed commands are labeled g_i, as shown in the figure. The g_i segments do *not* include the delta-dirac functions. Notice that the sum of all these elements (including the convolution with the delta-dirac functions) returns the original command. This process can be extended to the trolley response to any n-step, staircase command:

$$v_t(t) = \sum_{i=1}^{n} g_i(t) * \delta(t - t_i) \tag{8.7}$$

where v_t is the original trolley velocity, $g_i(t)$ are the pieces formed using the process illustrated in Figures 8.8, and t_i are the switch times.

Each of the $g_i(t)$ command segments share the following properties: they begin at t=0 and have zero value at this time, there is a "rise" period where the command is changing, after a finite amount of time the command remains fixed at a steady-state value. The steady-state crane response to each $g_i(t)$ can therefore be determined using the previous analysis for the single step response. Using (8.4) yields:

$$\begin{aligned} \theta^i_{ss}(t) &= \frac{|G_i(j\omega_n)|}{L} sin\Big(\omega_n t - \frac{\pi}{2} + \angle G_i(j\omega_n)\Big) \\ G_i(j\omega_n) &= G_i(\omega_n, A_1, \ldots, A_n, t_1, \ldots, t_n) \end{aligned} \tag{8.8}$$

where $\theta^i_{ss}(t)$ is the steady-state response to the i^{th} command segment $g_i(t)$. $G_i(s)$ is the Laplace transform of $g_i(t)$, and $G_i(j\omega_n)$ is the complex number that results from substituting $s = j\omega_n$ into $G_i(s)$. Notice that $G_i(j\omega_n$‘) is not only a function of the frequency, but the shaper parameters $A_1, \ldots, A_n$ and $t_1, \ldots, t_n$ as well. For the remainder of this chapter,

however, the shorter notation $G_i(j\omega_n)$ will be used.

The steady-state response of the crane to the entire command can now be obtained by substituting the expression for $\theta^i_{ss}(t)$ from (8.8) into $g_i(t)$ in (8.7):

$$\theta_{ss}(t) = \sum_{i=1}^{n} \frac{|G_i(j\omega_n)|}{L} sin\Big(\omega_n(t - t_i) - \frac{\pi}{2} + \angle G_i(j\omega_n)\Big) \tag{8.9}$$

Notice that each sine term is delayed by t_i. Using the vector notation developed in the previous section, the solution can be written in a more compact and revealing form:

$$\theta_{ss}(t) = Asin\left(\omega_n t + \phi\right) \tag{8.10}$$

$$A = \Big|\sum \vec{v}_i\Big| \qquad\qquad \phi = \angle\Big[\sum \vec{v}_i\Big] \tag{8.11}$$

$$\vec{v}_i = \frac{|G_i(j\omega_n)|}{L} \angle\Big[-\omega_n t_i - \frac{\pi}{2} + \angle G_i(j\omega_n)\Big] \tag{8.12}$$

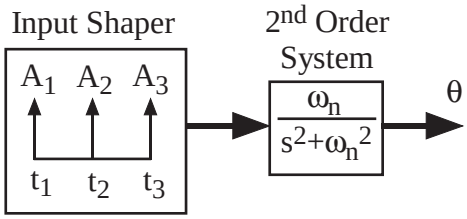
where the summation in (8.11) is understood to range from $i = 1 \ldots n$ and n is the number of impulses in the shaper.

The vectors, $\vec{v}_i$, carry important significance. Each one represents the steady-state vibration induced by the corresponding g_i piece of the original command. According to (8.10) and (8.11) these vectors can simply be added together to yield the amplitude and phase of the total response. This is nothing more than the principle of superposition. At first it may seem counterintuitive to use superposition for a nonlinear system. But recall in Figure 8.7 that the nonlinearity only exists in the drive, the planar crane block *is* linear. Therefore superposition *can* be used to find the deflection, θ, relative to the actual trolley velocity, v_t. This is the core idea for this chapter.

8.1.3 Insight into Vector Response

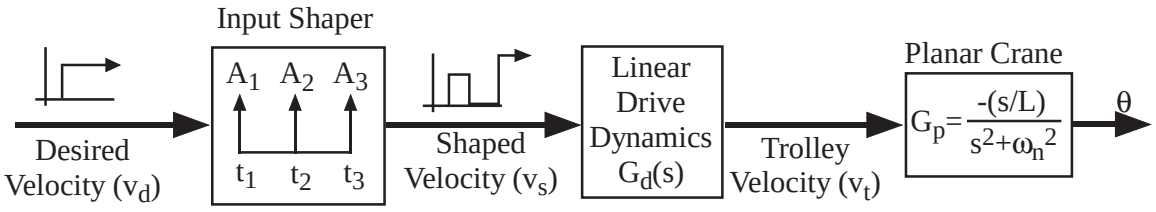
The last section concluded with a series of algebraic relations (8.10)-(8.12) for finding the steady-state response of the crane to a command. The goal is to use these relations to form shaped commands. To achieve this goal, three different systems using the same input shaper will be compared and contrasted:

Case I: A 2^{nd} order system commanded by input shaper impulses.



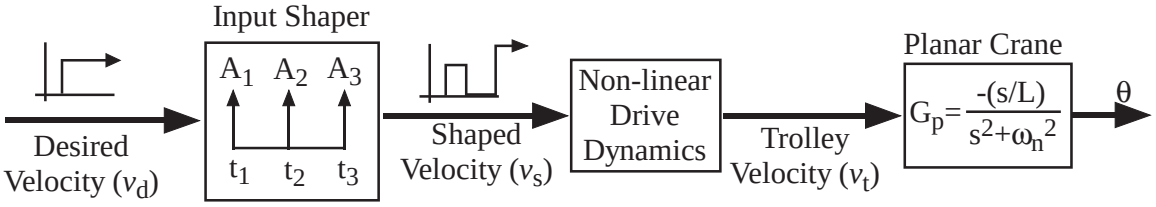
$$\vec{x}_i = |\tilde{A}_i| \angle \left[-\tilde{t}_i \omega_n + m_i \pi \right]$$

(a) Case I: A 2^{nd} Order System Commanded by Input Shaper Impulses.



$$\vec{y}_i = \tfrac{1}{L} |H(j\omega_n)| |A_i| \angle [-\omega_n t_i - \tfrac{\pi}{2} + \angle H(j\omega_n) + m_i \pi] \qquad \text{where,} \quad H(j\omega_n) = \left[\tfrac{v_f G_d(s)}{s} \right]_{s=j\omega_n}$$

(b) Case II: A Input-Shaped Crane Using a Linear Drive.



$$\vec{v}_i = \tfrac{1}{L} |G_i(j\omega_n)| \angle [-\omega t_i - \tfrac{\pi}{2} + \angle G_i(j\omega_n)]$$

(c) Case III: A Input-Shaped Crane Using a Nonlinear Drive.

Figure 8.9: Vector Analysis Applied to Three Cases.

Case II: A input-shaped crane using a linear drive.

Case III: A input-shaped crane using a nonlinear drive.

Each of these approaches is visually depicted in Figure 8.9. Each sub-figure contains a graphical description of the situation, as well as the vector representation of the residual vibration. The vectors $\vec{x}_i$, $\vec{y}_i$, and $\vec{v}_i$ give the steady-state response vectors (8.12) for each case. Notice that all the cases use the same 3-impulse shaper. However, the conclusions drawn from this example can be extended to any input shaper.

8.1.3.1 Case I: Input Shaper Response

Figure 8.9(a) shows a 2^{nd} order system commanded by the input shaper impulses directly. To find the steady-state response of the system, the input shaper command can be mathematically written in the Laplace domain:

$$I(s) = \sum_i A_i e^{-st_i} \tag{8.13}$$

The response of the 2^{nd}-order system is therefore:

$$\theta(s) = G_p(s) \cdot I(s) = \sum_i G_p(s) A_i e^{-st_i} \tag{8.14}$$

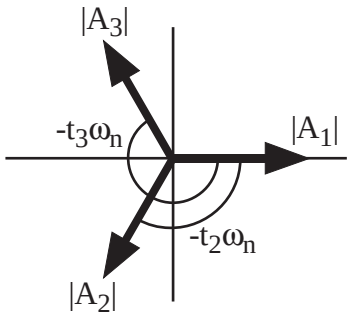
where $G_p(s) = \frac{\omega_n}{s^2+\omega_n^2}$ is the transfer function of the 2^{nd}-order system. Each term in the sum is the impulse response of the 2^{nd}-order system delayed by t_i. Equation (8.4) can be applied to find the steady-state response of each term. Then the terms can be summed together to yield the total steady-state response. In the interest of brevity, the steady-state response will be written using the vector notation discussed previously:

$$\vec{x}_i = |A_i| \angle \left[-\omega_n t_i + m_i \pi\right] \qquad \text{where,} \qquad m_i = \begin{cases} 0 & \text{if } A_i < 0 \\ 1 & \text{otherwise} \end{cases} \tag{8.15}$$

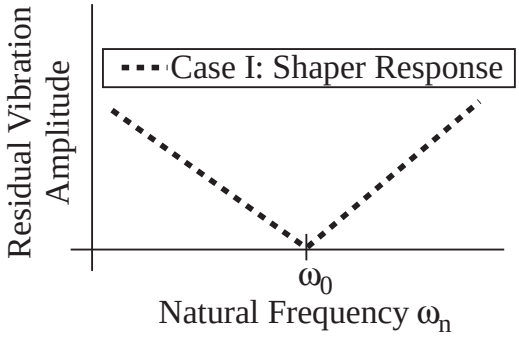
where $\vec{x}_i$ gives the amplitude and phase of the steady-state response to the i^{th} impulse. The purpose of the $m_i\pi$ term is to add π to the phase if A_i is negative. Equation 8.15 is also shown on the right-side of Figure 8.9(a). A polar plot of the $\vec{x}_i$ vectors is shown in Figure 8.10(a) for the 3-impulse shaper depicted in Figure 8.9(a).

The steady-state response to the total impulse sequence can be found using superposition: the steady-state responses to all the impulse are added together. Similarly, the $\vec{x}_i$-vectors can be added together and the resultant vector yields information about the steady-state response. The angle of the resultant gives the phase of the steady-state response, and the magnitude of the resultant gives the amplitude of the steady-state response. For the example shown in Figure 8.10(a), the vectors sum is zero, indicating that the steady-state response has zero vibration.

The above process can be repeated to find the steady-state response at different frequencies. According to (8.15), the angle of each vector will shift as the frequency varies.



(a) Vector Plot for Case I.



(b) Frequency Response (Sensitivity) for Case I.

Figure 8.10: Vector Diagram and Frequency Response for Case I.

As the vector angles shift, their sum will change. Figure 8.10(b) plots the magnitude of the vector sum (vertical axis) as a function of frequency (horizontal axis). The zero-vibration case from the previous paragraph is represented at the frequency $\omega_n = \omega_0$.

The vector approach for evaluating an input shaper's response comes directly from the literature [85]. It can also be used to design new input shapers: the shaper impulses are plotted as vectors, the vector geometry is manipulated so their sum yields the desired response (e.g. zero-vibration), and then the shaper parameters can be extracted from the plot. However, there are a couple differences between the "classic" vector-diagram approach (hereafter referred to simply as the vector-diagram approach) in the literature and the approach taken here. In the vector-diagram approach the $\omega_n t_i$ term for the phase is positive, whereas this term is negative in Figure 8.9(a). Vector diagrams are constructed this way because it is more visually and intuitively appealing for vectors that occur later in time, to have a positive, counter-clockwise angle. Equally valid vector diagrams could be constructed using a negative, clockwise angle. The second issue is that a $m_i\pi$ term appears in the phase in Figure 8.9(a), whereas the vector-diagram approach does not include this term. This is because the vector amplitudes are allowed to be negative in vector-diagrams. Neither of these two issues add new dynamics to the response.

8.1.3.2 Case II: Linear System Response

The second case to consider is an input-shaped crane with linear drive dynamics, shown in Figure 8.9(b). The transfer function of the drive is given by $G_d(s)$. The vector equation shown at the bottom of Figure 8.9(b) can be derived from (8.12).

The steady-state response of the linear system can be derived in the same manner as the previous case. Each $\vec{y}_i$ vector represents the steady state response of the i^{th} segment to the command. By superposition, the total steady-state response can be found by adding the steady-state response to each command segment. Similarly the sum of the $\vec{y}_i$ vectors yields the amplitude and phase of the steady-state response.

An interesting relationship exists between the shaper response vectors, $\vec{x}_i$, and the linear response vectors, $\vec{y}_i$. These vectors can be related mathematically as:

$$\vec{y}_i = \underbrace{\frac{1}{L}|H(j\omega_n)|}_{Global Scaling} \; |\vec{x}_i| \angle \left[\underbrace{-\frac{\pi}{2} + \angle H(j\omega_n)}_{Global Rotation} + \angle \vec{x}_i \right] \tag{8.16}$$

Notice that compared to the $\vec{x}_i$ vectors, the $\vec{y}_i$ vectors are all scaled by $\frac{1}{L}|H(j\omega_n)|$. This term is labeled "Global Scaling" in (8.16). In addition, the $\vec{y}_i$ vectors are all rotated by $-\frac{\pi}{2} + \angle H(j\omega_n)$ compared to the $\vec{x}_i$ vectors. This term is labeled "Global Rotation" in (8.16). The term global is used because all of the vectors are scaled and rotated by the same amount. Both of these terms can be traced back to the dynamic response of the linear drive and planar crane.

Figure 8.11 graphically depicts the effect of the global scaling and rotation terms as a geometric transformation. The transformation is broken up into two parts: scaling followed by rotation. The plot on the left of the figure shows the shaper response vectors $\vec{x}_i$. The middle plot shows the vectors after the global scaling transformation. The right plot shows vectors after both the global scaling and global rotation transformations which yields the linear response vectors $\vec{y}_i$.

The shaper response vectors sum to zero on the left of Figure 8.11. This indicates zero residual vibration, as discussed earlier. After the first scaling transformation, the vectors still sum to zero. This is always true since any number multiplied by zero is still zero.

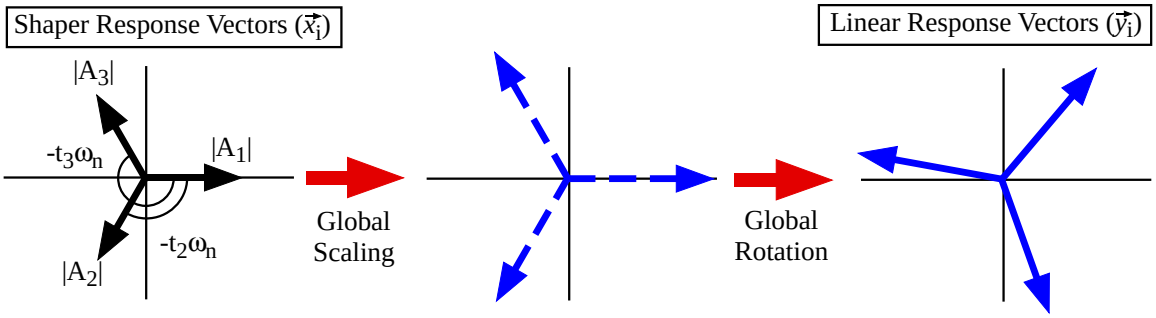


Figure 8.11: Relationship between Shaper Response Vectors ($\vec{x}_i$) and Linear Response Vectors ($\vec{y}_i$.)

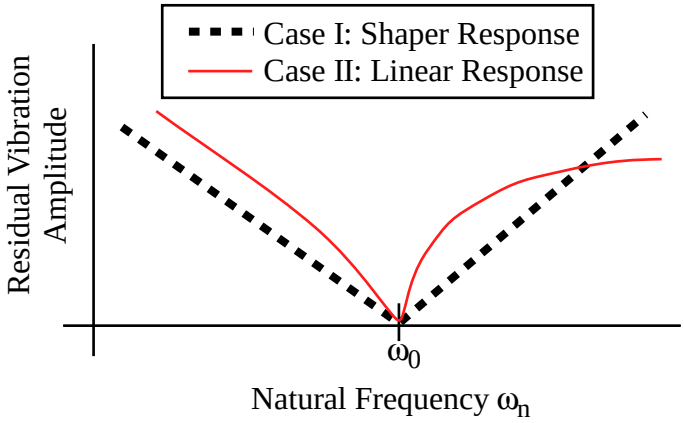


Figure 8.12: Frequency Response of Shaper and Linear System.

The second rotation transformation also does not change the sum since all the vectors are rotated the same amount. In fact, a global scaling and rotation transformation will never change the vector sum if it is zero. Therefore, if the shaper response vectors add to zero then the linear response vectors will always add to zero and have zero residual vibration. This is why a zero-vibration shaper works on any linear system, regardless of its dynamics.

The vector approach can also be used to find the residual vibration at other frequencies. As the frequency changes, the magnitude and angle of each vector in Figure 8.11 will change. This changes the vector sum which effects the residual vibration. Figure 8.12 shows a plot of the residual vibration amplitude verses frequency, similar to Figure 8.10(b). The dotted line is the shaper response curve shown earlier. The thin-solid line is the residual vibration of the linear system at each frequency. The zero-vibration case from the previous paragraph is shown at $\omega_n = \omega_0$. Notice that except at $\omega_n = \omega_0$, the two curves are different. This is

because the linear-response curve includes the effects of the global scaling and rotation. For non-zero vector sums, these transformations alter the amplitude of the vector sum. Recall that the reason for the global scaling and rotation was the added dynamic response of the drive and crane. These added dynamics are what causes the discrepancy between the two curves at non-zero values.

8.1.3.3 Case III: Nonlinear System Response

The third case to consider is the input shaped crane system with nonlinear dynamics, shown in Figure 8.9(c). The vectors $\vec{v}_i$, shown at the bottom of Figure 8.9(c), are derived directly from (8.12). Recall that each vector represents the steady-state amplitude and phase to a segment of the command. Once again, the steady-state response of total command can be found using superposition. Adding the $\vec{v}_i$ vectors yields amplitude and phase of the steady-state response to the total command.

The nonlinear response vectors, $\vec{v}_i$, differ from the linear vectors, $\vec{y}_i$, in two ways:

1. Earlier it was shown that the linear response vectors, $\vec{y}_i$, are related to the shaper response vectors, $\vec{x}_i$, by a global scaling and rotation transformation. The relationship between the nonlinear response vectors, $\vec{v}_i$, and the shaper response vectors, $\vec{x}_i$, is more complicated. Comparing the vector expressions in Figures 8.9(a) and 8.9(c) reveals that the transformation between each $\vec{x}_i$ vector and each $\vec{v}_i$ vector is different. In other words, there is a global *and local* scaling and rotation between the vectors: $|G_i(j\omega_n)|$ is the global and local scaling and $\angle G_i(j\omega_n)$ is the global and local rotation. The "local" terminology refers to the idea that each vector is effected differently. Mathematically it is represented by the i subscript on the $G_i(j\omega_n)$ terms.

2. The impulse amplitudes and times, A_i and t_i, form a nonlinear relationship with the amplitude and phase of the $\vec{v}_i$ vectors. Recall that in the linear system, the impulse amplitudes could be factored out of the amplitude of the $\vec{y}_i$ vectors. Furthermore, the impulse times were linearly related to the phase of the $\vec{y}_i$ vectors. This is no longer the case for the nonlinear system.

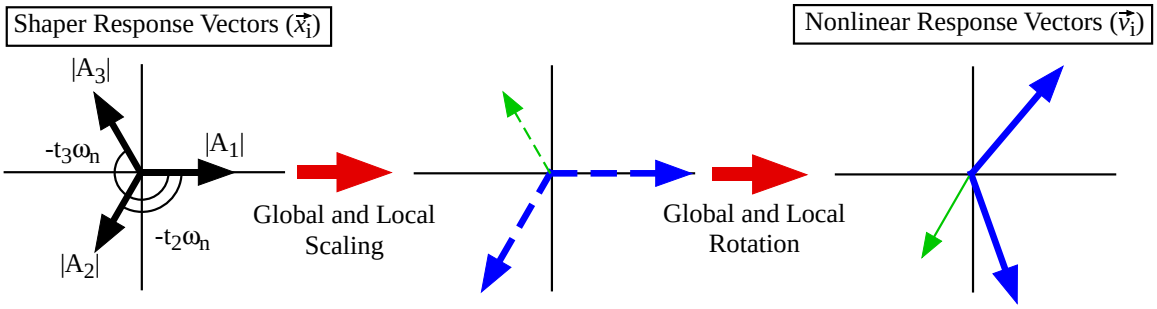


Figure 8.13: Relationship between Shaper Response Vectors ($\vec{x}_i$) and Nonlinear Response Vectors ($\vec{v}_i$.)

Figure 8.13 illustrates the relationship between the shaper response vectors, $\vec{x}_i$, and the nonlinear response vectors, $\vec{v}_i$. The plot on the left of the figure shows the $\vec{x}_i$ vectors. The transformation to the $\vec{v}_i$ vectors is broken into two steps: a scaling transformation, shown in the middle plot, followed by a rotation transformation. The plot on the right shows the $\vec{v}_i$ vectors. Notice that the middle vector, shown with the thin line, is scaled and rotated differently than the other two vectors. This represents the "local" aspect of the scaling and rotation transformation.

The sum of the $\vec{v}_i$ vectors yields the amplitude and phase of the steady-state response to the total nonlinear command. This rule can be applied to the example in Figure 8.13. Notice that although the $\vec{x}_i$ vectors add to zero, the $\vec{v}_i$ vectors do not add to zero. This implies that the command will have residual vibration. Earlier it was shown that if the shaper response vectors sum to zero, then their sum will be unaffected by the global scaling and shifting caused by the linear system dynamics. This is clearly no longer the case for the nonlinear system. Since each vector is scaled and rotated differently there is no guarantee that the $\vec{v}_i$ vectors will sum to zero if the $\vec{x}_i$ vectors sum to zero.

The residual vibration amplitude of the nonlinear system at different frequencies can also be compared to the shaper response (case I) and linear response (case II), as shown in Figure 8.14. The dotted line shows the shaper response, the solid line is the linear response, and the dashed line is the nonlinear response. Notice that at $\omega_n = \omega_0$ both the shaper response and linear response are zero. However, at this frequency the nonlinear response is non-zero, as explained in the previous paragraph. The residual vibration of the nonlinear

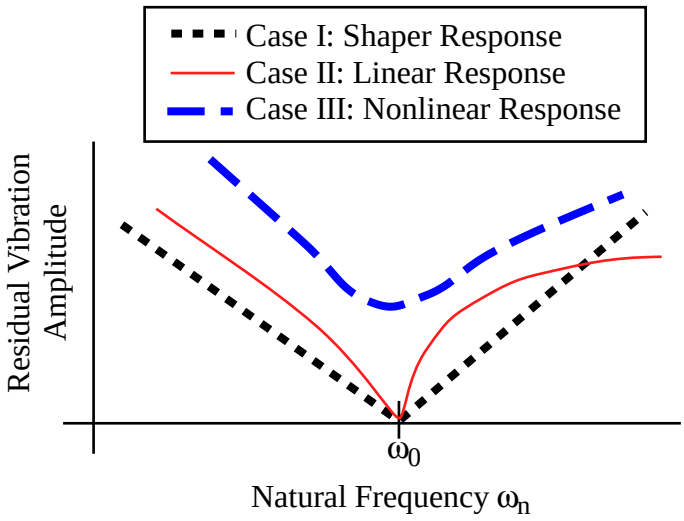


Figure 8.14: Frequency Response of Shaper, Linear System, and Nonlinear System.

system also changes with frequency in a totally different manner than the other two cases. As the frequency changes, the phase of each $\vec{v}_i$ vector changes due to the linear $-\omega_n t_i$ term and the nonlinear $\angle G_i(j\omega_n)$ term. In addition, the amplitude of each $\vec{v}_i$ vector changes with the frequency due to the nonlinear $|G_i(j\omega_n)|$ term. Recall that the nonlinear $G_i(j\omega_n)$ terms come from the nonlinear dynamics of the system. This is why the frequency response of the nonlinear system is different from the other two cases.

8.1.4 Zero-Vibration Shaper Approach

The above discussion can be used to form new strategies for creating shaped commands. The simplest strategy is to set the sum of the nonlinear response vectors, $\vec{v}_i$, equal to zero. This will force the residual vibration to be zero. Consequently, this shaping strategy is termed the *zero vibration approach*. Although the times and amplitudes don't appear explicitly the equations for $\vec{v}_i$, they are contained within the $G_i(j\omega_n)$ terms. So to implement this strategy one would first solve for $G_i(s)$ as a function of the shaper times and amplitudes. Then, substitute these functions into (8.12), set the sum equal to zero and solve for the shaper times and amplitudes.

The previous paragraph described an algebraic approach for creating a zero-vibration command. However, there is another approach. Because the steady-state amplitude and phase of vibration is given by $\sum \vec{v}_i$, these equations can be solved geometrically. All the

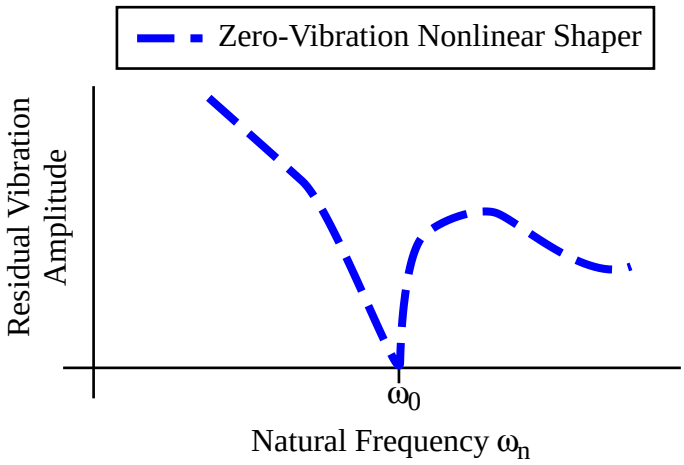


Figure 8.15: Frequency Response of Shaper, Linear System, and Zero-Vibration Nonlinear Shaper.

$\vec{v}_i$ vectors can be plotted on a polar plot, as previously shown in 8.6(a). Then geometric principles (eg. the law of cosines, the law of sines, etc...) can be used to determine the necessary relations that make the vectors sum to zero. Although there is no mathematical difference between the geometric and algebraic approaches, the geometric approach is usually less cumbersome in practice. A similar approach was used by Hekman and Singhose [25] to compensate for motor dynamics in cranes.

Figure 8.15 shows the frequency response of the nonlinear system using a shaper designed with the zero-vibration approach. The residual vibration is zero at the design frequency $\omega_n = \omega_0$. This is useful if the system frequency does not change. However, as the system frequency deviates from $\omega_n = \omega_0$ there is no way to predict how well the zero-vibration shaper will perform at different frequencies without simulating the system. In other words, the robustness of the shaper cannot be controlled. This is a major drawback of the zero-vibration approach.

8.1.5 Template Shaper Approach

A second shaping strategy, called the *template shaper approach*, also emerges from the preceding discussion. Recall that above discussion generated two main observations:

- Linear drive dynamics impart a global scaling and rotation on the response vectors.

- Nonlinear drive dynamics impart a global *and local* scaling and rotation on the response vectors.

The main idea behind the template shaper approach is to design the nonlinear shaper so it mimics the behavior of a pre-chosen linear shaper on a linear system. The pre-chosen linear shaper is called the *template shaper*. According to the above observations, this means that the nonlinear shaper must be designed to eliminate the local scaling and rotation of the response vectors. If such a shaper can be found, the template shaper response vectors, $\vec{x}_i$, will be related to the nonlinear response vectors, $\vec{v}_i$, solely by a global scaling and global rotation transformation. In this way the response of the nonlinear system mimics the behavior of a linear system.

Assuming that the aforementioned nonlinear shaper exists, the magnitude of all the $\vec{x}_i$ vectors and $\vec{v}_i$ vectors will be related by the same, global scaling factor:

$$\frac{|\vec{v}_i|}{Q} = |\vec{x}_i| \tag{8.17}$$

where Q is the global scaling factor. Notice that this scaling factor is the same for all i which is why it is called "global". In addition, assuming that the nonlinear shaper exists the angle of all the $\vec{x}_i$ vectors and $\vec{v}_i$ vectors will all be offset by the same, global rotation factor:

$$\angle\vec{v}_i - R = \angle\vec{x}_i \tag{8.18}$$

where R is the global rotation factor. Again, this rotation factor is the same for all i, which is why it is called "global". As the nonlinear drive dynamics approach a linear system, the Q and R constants approach the global scaling and rotation of the linear response vectors given in (8.16). If the drive dynamics become completely linear, then the solution for the nonlinear shaper converges to the template shaper. This is because the template shaper is a linear shaper and it will eliminate vibration on any linear system regardless of its dynamics. Note that these equations are formed at a single, model frequency. The solution is only valid at this frequency, and does not hold for other frequencies in general.

Additionally, the amplitudes, A_i, of the nonlinear shaper must be constrained to sum to one so the shaped command reaches the final desired velocity. The time of the first

impulse, t_1, must also be constrained to be zero. These constraints, together with the previous relations, form the equations for the template shaper solution:

$$\begin{aligned} \frac{|\vec{v}_i|}{Q} &= |\vec{x}_i| & \sum A_i &= 1 \\ \angle\vec{v}_i - R &= \angle x_i & t_1 &= 0 \end{aligned} \tag{8.19}$$

It is assumed that both the nonlinear shaper and template shaper use n impulses and $i = 1 \ldots n$ in the above equations. Notice that there are a total of 2n+2 equations: n equations relating the vector magnitudes, n equations relating the vector angles, and 2 constraint equations. Likewise there are 2n+2 unknowns: n shaper impulse amplitudes, n shaper impulse times, and the 2 constants Q and R.

These equations are put into final form by substituting the amplitude and phase for the $\vec{v}_i$ and $\vec{x}_i$ vectors and rearranging:

$$\begin{aligned} \frac{|G_i(j\omega_n)|}{LQ} &= |\tilde{A}_i| & \sum A_i &= 1 \\ t_i &= \tilde{t}_i - \frac{1}{\omega_n}\left(-\angle G_i(j\omega_n) + m_i\pi + R\right) & R &= \angle G_1(j\omega_n) - m_1\pi - \frac{\pi}{2} \end{aligned} \tag{8.20}$$

where $\tilde{A}_i$ and $\tilde{t}_i$ represent the impulse times and amplitudes of the template shaper. A_i and t_i represent the impulse times and amplitudes of the nonlinear shaper being solved for. Notice the value of R can be solved in closed form to satisfy the $t_1 = 0$ constraint.

Because of the way that the shaper template solution is formed, the nonlinear system response will mimic a linear system response. The nonlinear shaper formed by (8.19) ensures that the $\vec{v}_i$ vectors are related to the $\vec{x}_i$ vectors by a global scaling and rotation transformation at the model frequency. It was shown earlier that if a set of vectors sum to zero, their sum will remain zero after a global scaling and rotation transformation. It therefore follows that the nonlinear shaper will have zero residual vibration at the model frequency, if the template shaper response has zero residual vibration at that frequency. In this way, the performance of these nonlinear-template shapers is similar to the nonlinear-zero-vibration shapers discussed earlier. However, the nonlinear-template shapers have the added benefit that their frequency response is similar to the template shaper.

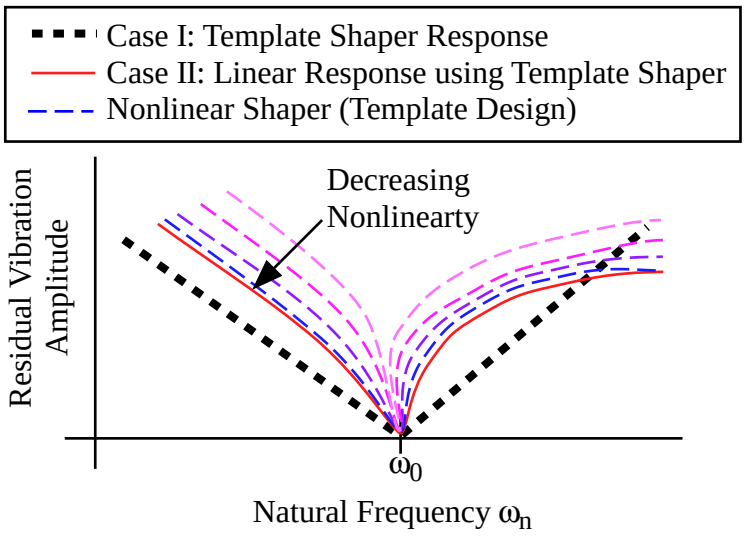


Figure 8.16: Frequency Response of Nonlinear Shaper (Template Design) for Various Levels Nonlinear Dynamic Behavior.

The main strength of using the shaper template approach is its frequency response. Figure 8.16 shows the residual vibration amplitude (vertical axis) verses frequency (horizontal axis). The dotted line is the $2^n d$-order system response to the template shaper, as shown earlier in Figure 8.9(a). The solid line is a linear system response to the template shaper, as shown earlier in Figure 8.9(b). The dashed lines show the response of the nonlinear system using a shaper designed with the shaper template approach. Each line represents a drive system with a different amount of nonlinear behavior. Notice that all of the dashed lines have zero-vibration at the model frequency $\omega_n = \omega_0$, as predicted in the previous paragraph. Also notice that as the drive's dynamics become more linear, the nonlinear frequency response becomes closer to the response of a linear system. This is the main strength of the template shaper technique: it finds a nonlinear shaper that approaches the frequency response of the template shaper on a linear system. In contrast, the aforementioned zero-vibration approach can only guarantee zero-vibration at the model frequency.

Since the drive system will have some amount of nonlinearity, there will always be some deviation between the nonlinear shaper frequency response and the desired template shaper frequency response. The amount of deviation will depend on the severity of the nonlinearity.

In addition, the deviation between the curves increases as the system frequency deviates from the model frequency.

Equations (8.20) can be used to find the times and amplitudes for a nonlinear shaper. However, the difficulty in solving these equations depends entirely on the nature of the nonlinearity. In the best scenario, the equations are all uncoupled. Mathematically, this condition can be stated as $G_i(j\omega_n) = G_i(\omega_n, A_i, t_i)$. Physically, this means that each shaper time, t_i, and amplitude, A_i, only effects the response of the corresponding segment $g_i(t)$. If the equations are coupled the solution can be much more difficult to find. In some cases there may be no solution.

There are two scenarios in which a solution is guaranteed. The first is the trivial case that the drive block is linear. In this case the shaper times and amplitudes are equivalent to the shaper template: $A_i = \tilde{A}_i$ and $t_i = \tilde{t}_i$. The second case is if the nonlinearity is static. In this case the relationship between the input and output of the nonlinear drive block in Figure 8.7 can be written as $v_t = f(v_s)$. If a n-step shaped command is used for the shaped command, v_s, the resulting trolley velocity, v_t, is also an n-step profile where n is equal to the number of impulses in both the nonlinear shaper and the shaper template. The $G_i(j\omega)$ values can be written as:

$$G_i(j\omega) = \frac{v_i - v_{i-1}}{j\omega} \tag{8.21}$$

where v_i is the velocity of the i^{th} step of the trolley velocity v_t. In addition, $v_0 = 0$ and $v_n = v_f$ where v_f is the trolley final velocity. Substituting (8.21) into the nonlinear shaper solution (8.20) yields:

$$\begin{aligned} t_i &= \tilde{t}_i \\ v_i &= v_f \sum_{1}^{i} \tilde{A}_i \end{aligned} \tag{8.22}$$

The static-nonlinearity relation, $v_t = f(v_s)$, can then be used to solve for the shaped command impulse amplitudes A_i given the v_i velocity step values. This assumes that $f^{-1}(v_i)$ exists for all i.

Future work could extend the scope of this solution by adding more impulses. With these added degrees of freedom, the frequency response of the nonlinear shaper and the

template shaper could be set equal at multiple frequencies rather than just a single design frequency. The result would be an even more accurate reconstruction of the template frequency response for the nonlinear shaper.

8.1.6 Summary of the Vector Approach for Continuous Systems

This section has presented two analytical ways to find input-shaped commands for nonlinear systems. Both approaches begin by finding the vector response of the system. The steps for finding the vector response are:

1. Divide the original command into segments $g_i(t)$.

2. Find the Laplace Transform of each segment $G_i(s)$ as a function of the shaper times and amplitudes.

3. Substitute $G_i(s)$ into (8.12) to find the steady-state response of each $g_i(t)$ segment. This is represented by the $\vec{v}_i$ vectors.

Additionally, the sum of the $\vec{v}_i$ vectors yield the amplitude and phase of the steady-state response to the whole command.

The first approach for finding a shaped command is to simply set $\sum \vec{v}_i = 0$ and solve for the shaper times and amplitudes. This approach will be used to find UMZV commands for a system for a braking nonlinearity in the next section.

The second approach for finding a shaped command uses a linear shaper as a template. This solution appeared in (8.19). The idea is to relate the template shaper vectors, $\vec{x}_i$, to the nonlinear response vectors, $\vec{v}_i$, via a global scaling factor Q and a global rotation R. The result is that the frequency response of the nonlinear system using the compensated command mimics the frequency response of a linear system using the template shaper.

8.2 Application: Braking Nonlinearity

This section focuses on non-symmetrical acceleration-braking, as illustrated in Figure 8.17. The desired command is assumed to be a pulse with duration t_p. In the figure the solid

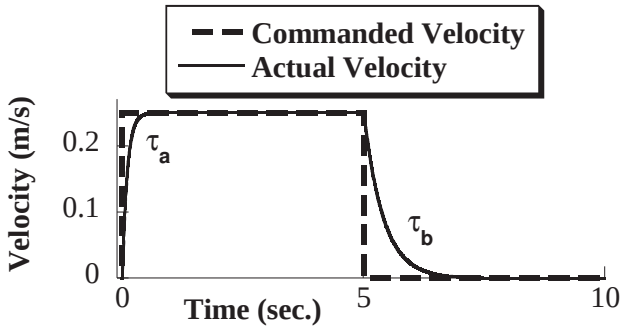


Figure 8.17: Non-Symmetrical Acceleration-Braking.

line shows the desired pulse command, and the pulse duration is 5 sec. The velocity accelerates up to the setpoint speed using an exponential rise with time constant τ_a. However, the system brakes with a different time constant, τ_b. Such a nonlinearity would occur in systems using a clutch or some other mechanical element for braking. It can also occur because the electrical circuitry responds differently to acceleration and braking. The question arises: *how do standard, linear input shapers perform in the presence of the non-symmetrical acceleration-braking shown in Figure 8.17?*

First, the system and experimental setup will be described. Then, the performance of standard ZV and UMZV shaper will be examined. It will be demonstrated that the braking nonlinearity effects the UMZV shaper more than the ZV shaper. The concepts from the previous section will then be used to form a new type of UMZV command that compensates for the non-symmetrical acceleration-braking. The performance and limitations of this new command are then discussed. The theoretical results will be supported by simulations and experimental results.

8.2.1 System Definition and Experimental Setup

A typical implementation of an input shaper is given in Figure 8.18. This figure includes the nonlinear braking element discussed earlier. The unshaped command, $r(t)$, is a pulse with time duration t_{p}. This signal passes through an input shaper to form the shaped command, $r_{\mathrm{s}}(\mathrm{t})$. The drive system cannot follow the shaped command exactly due to the nonlinear braking effect. This block has an acceleration time constant τ_a and a braking time constant τ_b, as shown in Figure 8.17. While there have been some studies about designing input

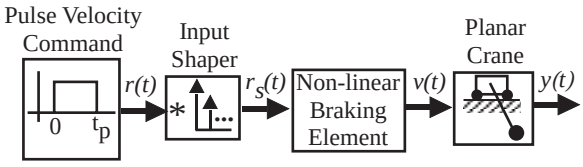


Figure 8.18: Crane Block Diagram.

Table 8.1: Bridge Crane Experimental Parameters.

Setpoint Velocity $[v_{max}]$	Suspension Length $[L]$	Acceleration Constant $[\tau_a]$
0.17 (m/s)	0.84 (m)	0.117 (s)

shapers for nonlinear systems [40, 45, 70, 82] none have focused on this particular type of nonlinearity.

The control architecture shown in Figure 8.18 was implemented on the portable bridge crane. In this case the non-symmetrical acceleration-braking was programmed into the PLO. The hanging payload deflection was recorded with a digital camera. The configuration parameters used for all of the experiments in this section are shown in Table 8.1.

8.2.2 ZV Shaping with a Braking Nonlinearity

Previously it was shown that a ZV shaper will eliminate the residual vibration for a linear system. So the question arises: *how will ZV shaping be effected by the nonlinear braking element?* It will be shown that the ZV shaper works well in spite of the nonlinear braking effect.

Suppose a ZV shaper were implemented in the input shaper block in Figure 8.18. As the pulse duration of the velocity command varies, the structure of the ZV shaped command changes. This, in turn, effects how successful the ZV shaper is at eliminating vibration. To clarify this issue, a shaped command will be categorized as either a *short command*, *long command*, or *interference command* depending on the value of the pulse duration relative to the shaper duration. Figures 8.19(a), 8.19(b), and 8.19(c) show each command respectively.

8.2.2.1 ZV Short Commands

Figure 8.19(a) shows an example of a ZV shaped *short command*, as well as the resulting velocity. It consists of two pulses. The key feature is that the velocity returns to zero before

the second pulse in the shaped command. In order for this to occur the following constraint on the pulse duration (t_p) must be satisfied:

$$t_p < t_2 - 3\tau_b \tag{8.23}$$

where $t_2 = \frac{T}{2}$ is the time of the second impulse of a ZV shaper. Note that $3\tau_b$ is used as the approximate time for the exponential decay to reach zero (steady state). For some systems with a long τ_b, *short commands* may not exist.

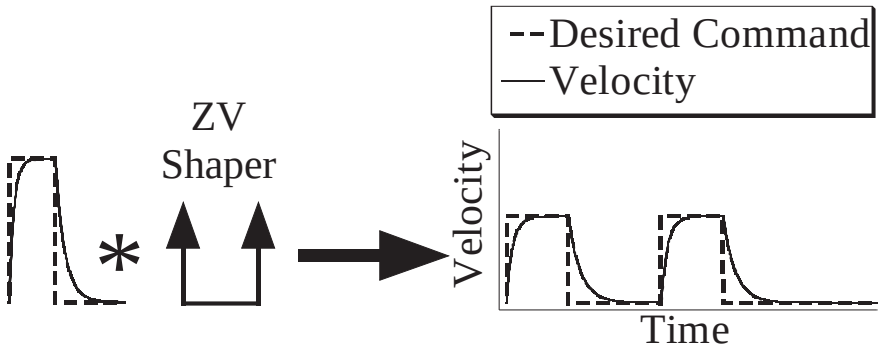
Figure 8.19(a) shows how a ZV *short command* is the convolution of a ZV shaper with a pulse. The velocity response can also be formed as the convolution of a ZV shaper with a smoothed pulse despite the nonlinear braking. This smoothed pulse is formed by applying the non-symmetrical acceleration-braking to the original pulse command. Note that this decomposition is only possible when assumption (8.23) holds. Because a ZV shaper can be deconvolved from the resulting velocity, the command will yield zero residual vibration.

8.2.2.2 ZV Long Commands

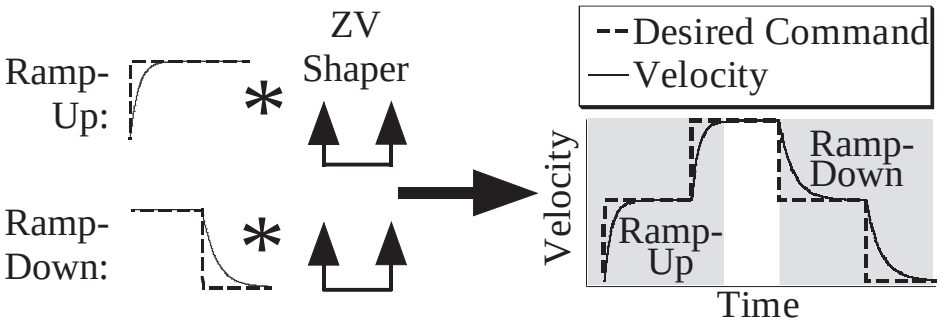
Figure 8.19(b) shows an example of a ZV shaped long command and the resulting velocity. It consists of a two-step ramp-up segment, a coasting segment, and a two-step ramp-down segment. The key feature of this type of command is that the velocity reaches its full speed before the ramp-down segment. This condition yields the following constraint on pulse duration (t_p):

$$t_p < t_2 + 3\tau_a \tag{8.24}$$

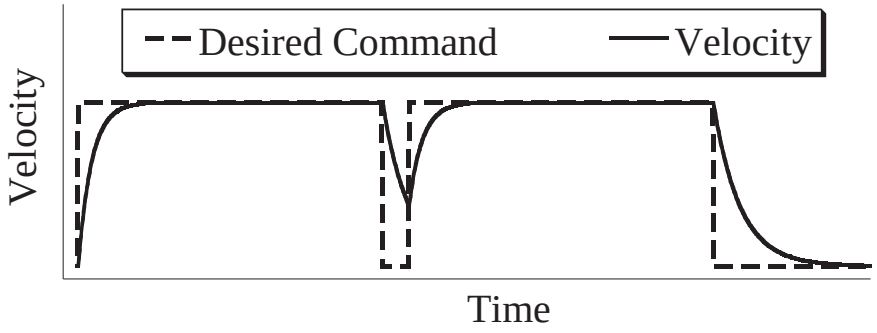
Figure 8.19(b) also shows how long commands can be decomposed into two parts. Suppose the command were divided into ramp-up and ramp-down segments. The ramp-up command can be expressed as the convolution of a step and a ZV shaper. Similarly, the velocity profile can be expressed as the convolution of a exponential step with a ZV shaper. The smoothed step is formed by applying the non-symmetrical acceleration-braking to the original step. Again, this decomposition is only possible when assumption (8.24) holds. As a result, the ramp-up segment will cause zero residual vibration. Similar arguments can be made about the ramp-down segment. Because the total command can be formed as the sum of these two segments, a ZV shaped *long command* will cause zero residual vibration.



(a) Deconvolution of a *ZV short command.*



(b) Deconvolution of a *ZV long command.*



(c) *ZV Interference Command.*

Figure 8.19: Short, Long, and Interference ZV Commands.

8.2.2.3 ZV Interference Commands

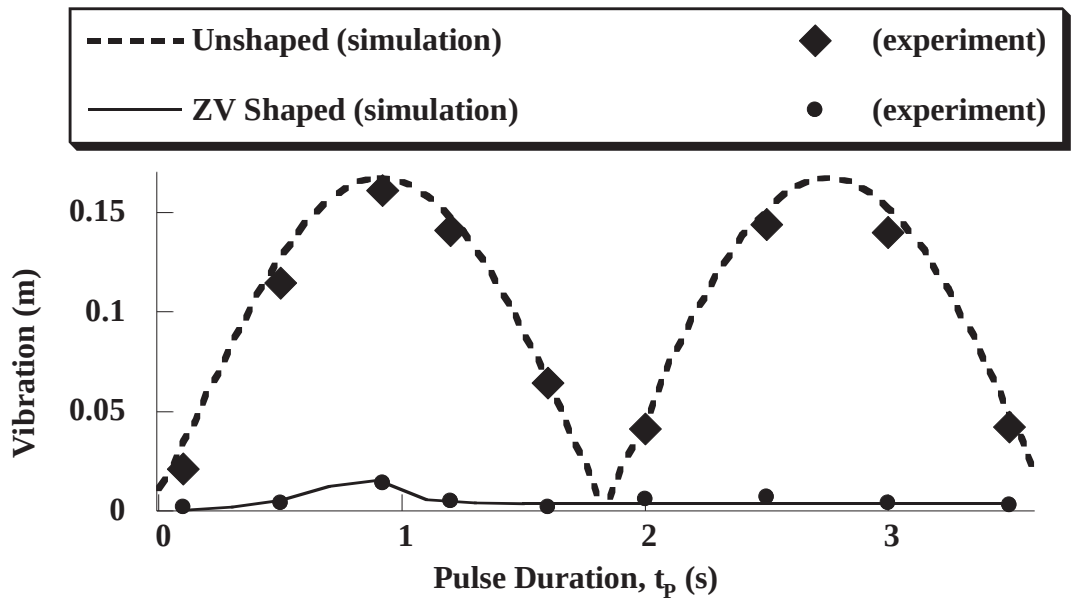
In this case, a change in the command interferes with the exponential acceleration or braking process. Figure 8.19(c) shows such an example. This scenario occurs when the pulse duration is between a short and long command. Because of its structure, a ZV shaper cannot be deconvolved out of an *interference command*. Therefore, ZV shaped *interference commands* will yield some residual vibration.

To test the conclusions made above, both simulations and experiments were conducted. Figure 8.20(a) compares the residual vibration of ZV shaped commands to unshaped commands. The horizontal axis is pulse duration and the vertical axis is the vibration induced by the corresponding command. Notice that all of the ZV-shaped commands out perform unshaped commands. Figure 8.20(b) shows a close-up view of the ZV-shaped results. The figure divides the commands into *short* (I), *interference* (II), and *long* (III) using (8.23) and (8.24). Both *short* and *long commands* have very low residual vibration, while *interference commands* cause substantially more residual vibration.

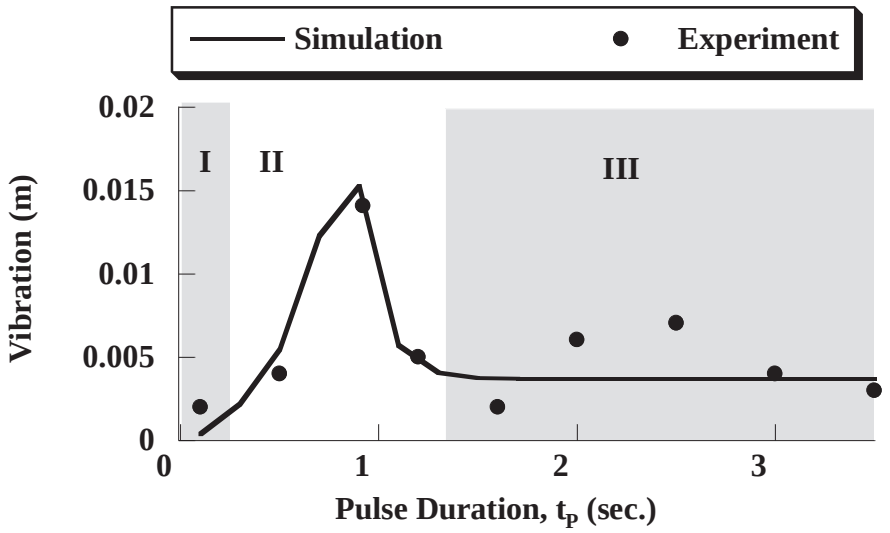
Two important points should be made about the results. First of all, the range of *long commands* extends infinitely outward along the horizontal axis in Figure 8.20(b). So, for a large range of pulse durations, a ZV shaped command performs very well. The second point is that even though *interference commands* can cause residual vibration, it is still small compared to the residual vibration induced by unshaped commands.

8.2.2.4 Effect of Time Constants

As the acceleration and braking time constants vary, the severity of the braking nonlinearity changes. In particular, as $\tau_a \rightarrow \tau_b$ the system behaves more like a linear system and a ZV shaper becomes more effective. Figure 8.21 shows the effect of changing both the pulse duration and the braking time constant. The vertical axis shows the vibration for each case. The figure plots both experimental and simulated results. Each line is analogous to the data presented in Figure 8.20. Notice that since τ_a is fixed at 0.117 (see Table 8.1), as $\tau_b \rightarrow \tau_a = 0.117$ the overall vibration is at a minimum because the braking nonlinearity vanishes.



(a) ZV vs. Unshaped



(b) ZV Performance

Figure 8.20: Residual Vibration using Various Pulse Durations $[\tau_b = 0.065\ (s)]$.

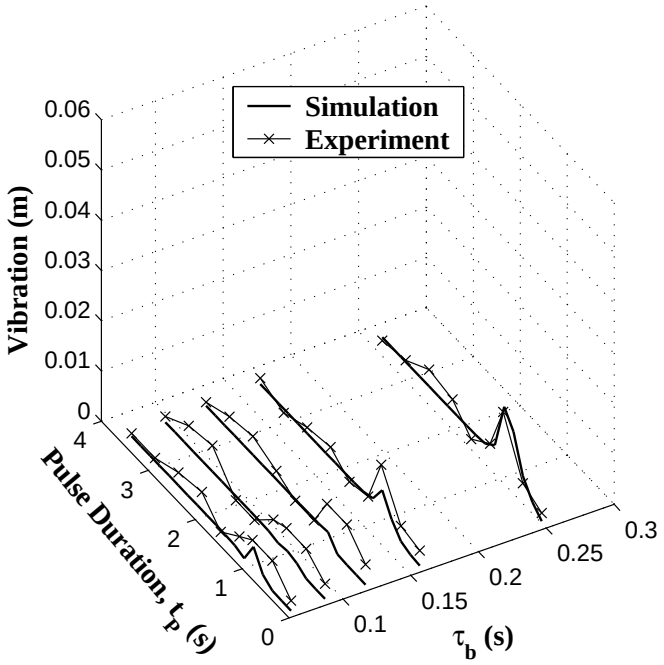


Figure 8.21: Vibration of ZV Commands for Various t_p and τ_b.

8.2.3 UMZV Shaping with a Braking Nonlinearity

A UMZV shaper has some advantages over a ZV shaper. The shaper yields faster responses and can be implemented with on/off type drive systems. Therefore, a natural question to ask is: *how is a UMZV shaper affected by the braking nonlinearity under discussion?* The previous section demonstrated that despite the non-symmetrical acceleration-braking, ZV shaping still performed well under most conditions. However, this section will show that a UMZV shaper will have degraded performance due to the nonlinearity over a wide range of parameters.

When discussing the effects of pulse duration it is again useful to categorize the commands as *short*, *long*, or *interference*.

8.2.3.1 UMZV Short Commands

Figure 8.22 shows an example of a UMZV short command and the resulting velocity. Equation (8.23) can also be used as the constraint on t_p for UMZV short commands. However, in this case $t_2 = \frac{T}{6}$ is the time of the second impulse of a UMZV shaper. Figure 8.22 also shows how a UMZV *short command* can be expressed as the convolution of a UMZV shaper

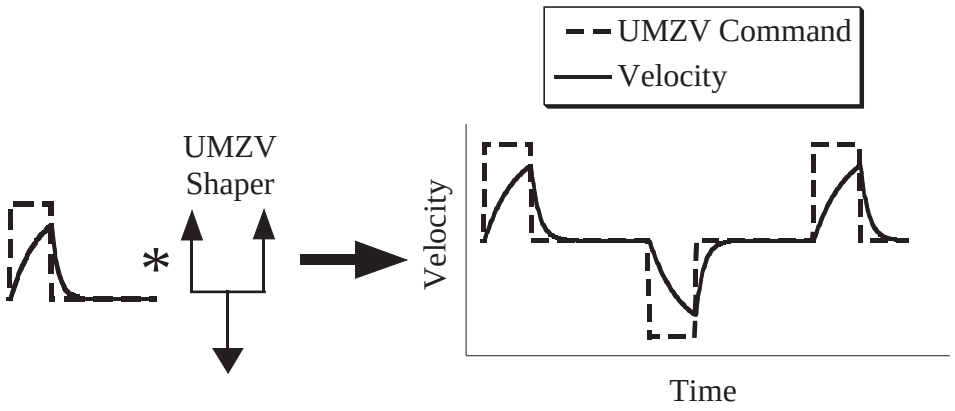


Figure 8.22: Deconvolution of a UMZV Shaped *Short Command*.

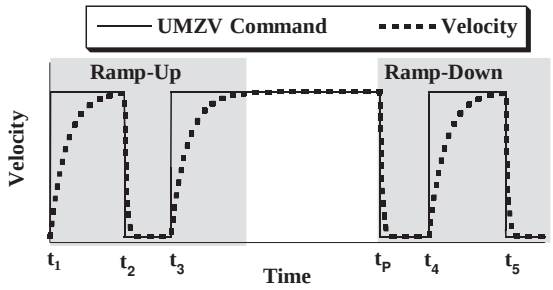


Figure 8.23: UMZV_C Command Template.

with a pulse. Similar to Section 8.2.2.1, the velocity can also be written as the convolution of a UMZV shaper with a smoothed pulse. Because this deconvolution is always possible, a UMZV short command will not induce residual vibration.

8.2.3.2 UMZV Long Commands

Figure 8.23 shows an example of a UMZV *long command* along with the resulting velocity. Using the same definition given in Section 8.2.2.2, a constraint on the pulse duration can be formed:

$$t_p > t_3 + 3\tau_a \tag{8.25}$$

where $t_3 = \frac{T}{3}$ is the time of the third impulse of a UMZV shaper. However, unlike ZV *long commands*, a UMZV shaper cannot be deconvolved out of a UMZV *long command*.

For the short command the deconvolution could be done by inspection because the velocity profile could be divided into three equivalent segments with alternating sign. The UMZV shaper is composed of three equivalent impulses with alternating sign, thereby suggesting the deconvolution in Figure 8.22. For the long command the three segments would be the rising-falling-rising portions of the ramp-up command. Since these are not equivalent the UMZV shaper cannot be de-convolved out. So, in general, a UMZV *long command* will cause some residual vibration.

8.2.3.3 UMZV Interference Commands

An interference command falls in between a short and long command. The same arguments presented in Section 8.2.2.3 also hold here. Therefore, an interference command will induce residual vibration.

To test the above conclusions, both simulations and experiments were performed. Figure 8.24 shows residual vibration as pulse duration is varied for unshaped and UMZV shaped commands. The experimental data was taken from the portable bridge crane discussed earlier. The figure divides the graph into *short* (I), *interference* (II), and *long* (III) commands. The figure shows that the system has measurable residual vibration for both interference and long commands. However, for the particular values of $\tau_a$and $\tau_b$used, the vibration from *interference commands* is fairly small. Comparing Figures 8.24 and 8.20(a), the range of commands that cause residual vibration for UMZV commands is much larger than for ZV commands. However, the UMZV shaped commands still have much less vibration, on average, than the unshaped commands.

8.2.3.4 UMZV Commands: Effect of Time Constants.

The same conclusions presented in Section 8.2.2.4 regarding the acceleration and braking constants, can be drawn here. Namely, as $\tau_b \rightarrow \tau_a$ the system behaves more like a linear system. As a result, the effectiveness of the UMZV shaper improves under these conditions. Figure 8.25 shows the effect of changing both the pulse duration and the braking time constant. The vertical axis shows the vibration for each case. Notice that as $\tau_b \rightarrow \tau_a = 0.117$ the overall vibration approaches zero. However, notice that over most of the operating space,

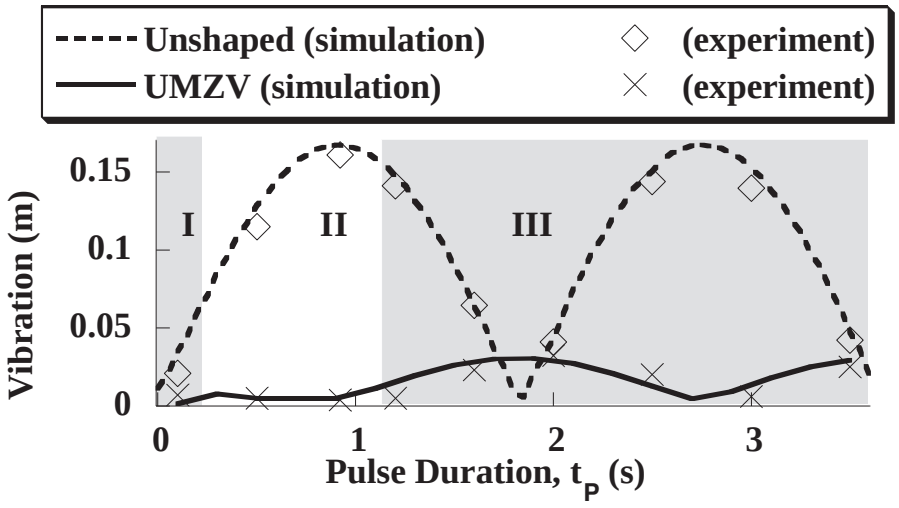


Figure 8.24: Vibration of Unshaped and UMZV Shaped Pulse for Various Pulse Durations $[\tau_b = 0.065(s)]$.

UMZV-shaped commands induce noticeable residual vibration, unlike the ZV commands shown in Figure 8.21.

8.2.4 Formulation and Implementation of a UMZV$_C$ Shaped Command.

As mentioned earlier, a UMZV shaper has the advantages of being fast and compatible with on/off actuators. However the last section showed that the effectiveness of UMZV shapers are degraded by the braking nonlinearity. This section will show how to formulate a new, modified UMZV shaper that retains the advantages of the standard shaper, while compensating for the braking nonlinearity. The new version of this shaper will be referred to as UMZV$_C$ (the "c" stands for compensated). The technique used to find this new shaper utilizes the zero-vibration vector solution presented in Section 8.1.

Figure 8.23 can be used to graphically represent the problem at hand. The goal is to find times $t_1, \ldots, t_5$ such that the command will yield zero residual vibration. Note that the command shown in Figure 8.23 would be classified as a long command according to the definition in Section 8.2.3.2. The first step is to find the residual vibration of the ramp-up portion as a function of times t_1, t_2, and t_3 and set it equal to zero. Then, a similar procedure can be applied to the ramp down portion.

Following the technique developed in Section 8.1, the trolley velocity, $v_t(t)$, is divided

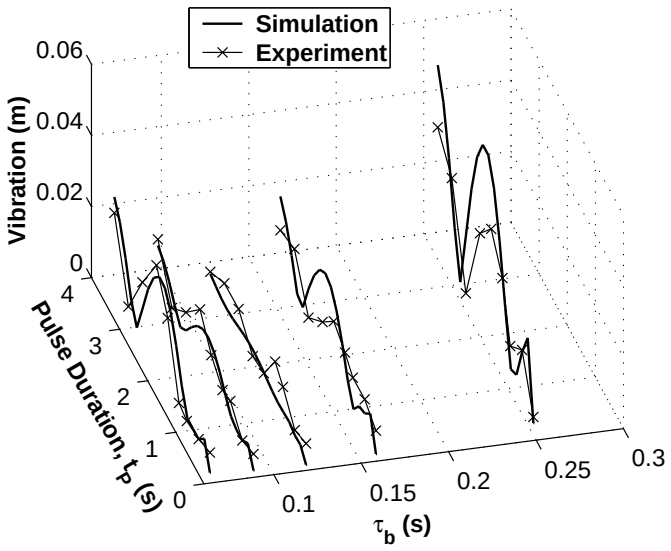


Figure 8.25: Vibration of UMZV Commands for Various t_p and τ_b.

into three segments, $g_i(t)$:

$$v_t(t) = \sum_{\mathrm{i}=1}^{3} g_i(t) * \delta\left(t - t_i\right) \tag{8.26}$$

where,

$$g_1(t) = v_{max} \cdot \left(1 - e^{-t/\tau_a}\right) \tag{8.27}$$

$$g_2(t) = -v_{max} \cdot \left(1 - e^{-t/\tau_b}\right) \tag{8.28}$$

$$g_3(t) = v_{max} \cdot \left(1 - e^{-t/\tau_a}\right) \tag{8.29}$$

where v_{max} is the maximum velocity. Each segment, $g_i(t)$, consists of a 1^{st}-order response to a step command.

The next step is to convert the $g_i(t)$ segments into the Laplace domain. Then, this is substituted into (8.10)-(8.12) to find the steady-state response, θ_{ss}:

$$\theta_{ss}(t) = Asin\left(\omega_n t + \phi\right) \tag{8.30}$$

$$A = \left|\sum \vec{v}_i\right| \qquad\qquad \phi = \angle\left[\sum \vec{v}_i\right] \tag{8.31}$$

$$\vec{v}_1 = \frac{v_{max}}{L\sqrt{(\tau_a\omega)^2+1}} \angle \left[tan^{-1}\left(\frac{1}{\tau_a\omega}\right) - \frac{3\pi}{2} \right]$$
$$\vec{v}_2 = \frac{v_{max}}{L\sqrt{(\tau_b\omega)^2+1}} \angle \left[tan^{-1}\left(\frac{1}{\tau_b\omega}\right) - \frac{\pi}{2} \right] \tag{8.32}$$
$$\vec{v}_3 = \vec{v}_1$$

Recall that the $\vec{v}_i$ vectors represent the amplitude and phase of the steady-state response to each $g_i(t)$ segment of the command.

Because the goal is to make (8.30) equal zero, the vectors (8.32) can be scaled, rotated, or reflected across the real-axis without affecting the results. Using these properties, a simplified set of vibration vectors can be formed:

$$\vec{v}_1 = 1 \angle[0]$$
$$\vec{v}_2 = \sqrt{\frac{(\tau_a\omega)^2+1}{(\tau_b\omega)^2+1}} \angle \left[\omega t_2 + \tan^{-1}\left(\frac{1}{\omega\tau_a}\right) - \tan^{-1}\left(\frac{1}{\omega\tau_b}\right) \right] \tag{8.33}$$
$$\vec{v}_3 = 1 \angle[0]$$

Note that here we have used the assumption that time t_1 will be zero.

There are two methods for finding times t_2 and t_3 that yield zero residual vibration. One method is to substitute (8.33) into (8.30), set the result equal to zero, and solve algebraically. While this method will yield a solution, the calculations become tedious. A much more efficient, and insightful, method is to use a geometric approach. Each of the three vectors (8.32) are shown in Figure 8.26. For the vectors to sum to zero they must form a triangle. Note that vector $\vec{v}_3$ is translated from quadrant II (solid) to quadrant I (dashed), in order to form a triangle.

Figure 8.26 also shows how the angles of the triangle relate to the vector angles given in (8.33). Note that $\alpha_2 = \angle\vec{v_2}$ and $\alpha_3 = \angle\vec{v_3}$. With the vectors arranged in a triangle, the law of cosines can be used to solve for the angles α_2 and α_3. Then (8.33) can be used to

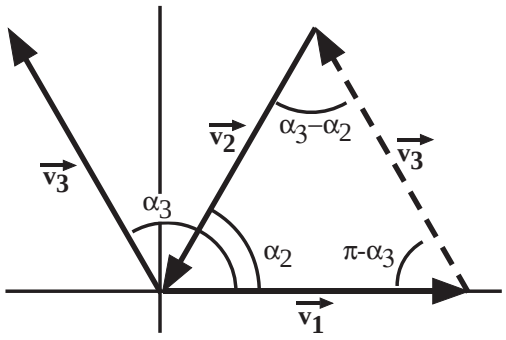


Figure 8.26: Phaser Diagram.

solve for times t_2 and t_3:

$$\begin{aligned} t_2 =& 1/\omega \left(\tan^{-1}\left(\frac{1}{\omega\tau_b}\right) - \tan^{-1}\left(\frac{1}{\omega\tau_a}\right)\right) + 1/\omega \cos^{-1}(\beta) + nT \\ t_3 =& 1/\omega \cos^{-1}\left(2\beta^2 - 1\right) + mT \\ \beta =& \frac{1}{2}\sqrt{\frac{(\tau_a\omega)^2 + 1}{(\tau_b\omega)^2 + 1}} \end{aligned} \tag{8.34}$$

where n, m are positive integers and $n \leq m$ to ensure that $t_2 < t_3$.

A few comments should be made about the solution given in (8.34):

- The duration of the command, t_3, will always be less than $\frac{T}{2}$ provided $n = m = 0$. This means that the UMZV$_\mathrm{C}$ command will always be shorter than a ZV command if $n = m = 0$.

- Additional solutions can be found by adding integer multiples of the system period, T, to t_2 or t_3. This appears as the mT and nT terms in (8.34).

- The β argument of the $\cos^{-1}$ terms in (8.34) must be between $[-1,\ 1]$ for the result to be a real number, thereby constraining the values of $\omega\tau_a$ and $\omega\tau_b$. The physical interpretation of this restriction can be illustrated using Figure 8.23. The vibration induced by the steps at times t_1 and t_3 (accelerating) must be canceled by the vibration from the step at time t_2 (braking). However, if $\tau_b \ll \tau_a$, then the vibration from the t_1 and t_3 steps is very small compared to the vibration from the step at t_2. In this case, the vibration cannot be canceled, no matter how t_2 and t_3 are chosen. Under these conditions, the β term is outside the range $[-1,\ 1]$ and no real solution can be found.

A UMZV$_{\mathrm{C}}$ command cannot be formed by straightforward convolution of a pulse with an input shaper. More specifically, note that $t_4 \neq t_p + t_2$ and $t_5 \neq t_p + t_3$. The reason is that during the ramp-up portion the velocity will: accelerate-brake-accelerate. However, during the ramp-down process this sequence is: brake-accelerate-brake. Because the sequences are different, the command switch-times must be different. To correctly solve for the ramp-down times, switch the values of τ_a and τ_b in (8.34) and solve for new times $\bar{t_2}$ and $\bar{t_3}$. Then solve for t_4 and t_5 using:

$$t_4 = t_p + \bar{t_2} \qquad t_5 = t_p + \bar{t_3} \tag{8.35}$$

8.2.5 Evaluation of UMZV$_{\mathrm{C}}$ Commands

The UMZV$_{\mathrm{C}}$ command retains the main advantages of a standard UMZV command: it is faster than ZV shaped commands, and uses only on/off commands. This section will examine the benefits and limitations of the UMZV$_{\mathrm{C}}$ command. It will be shown that UMZV$_{\mathrm{C}}$ commands are more effective than standard UMZV commands at reducing vibration over a wide range of parameters.

8.2.5.1 UMZV$_C$ Commands: Effect of Time Constants

Changing the acceleration and braking time constants affects the functionality of the UMZV$_{\mathrm{C}}$ shaped commands, although not nearly as much as standard UMZV commands. As was the case earlier, when $\tau_a = \tau_b$ the system is linear. The UMZV$_{\mathrm{C}}$ command becomes equivalent to a UMZV command and there is zero residual vibration. However, there are other effects that must be considered.

The derivation of the UMZV$_{\mathrm{C}}$ command made several assumptions about the values of τ_a and τ_b. Notice that in Figure 8.23 the system reaches the commanded velocity before the next switch time. These constraints can be mathematically stated as:

$$\begin{aligned} c_1: &\quad t_2 > 3\tau_a, &\quad t_4 - t_p > 3\tau_b \\ c_2: &\quad t_3 - t_2 > 3\tau_b, &\quad t_5 - t_4 > 3\tau_a \end{aligned} \tag{8.36}$$

where the constraints are labeled c_1 and c_2 respectively.

Another restriction on the acceleration and braking constants comes from the cos^{-1} terms in (8.34). As discussed earlier, the arguments of these terms must lie between $[-1, 1]$.

This feasibility constraint can be mathematically stated as:

$$\omega\tau_b > 0.5\sqrt{(\omega\tau_a)^2 - 3} \qquad\qquad \omega\tau_a > 0.5\sqrt{(\omega\tau_b)^2 - 3} \tag{8.37}$$

for both ramp-up and ramp-down segments in Figure 8.23.

8.2.5.2 UMZV$_C$ Commands: Design Algorithm

To create a UMZV$_C$ command one must evaluate equations (8.34) and (8.35) and then possibly adjust the m and n parameters to satisfy the constraints in (8.36). One also needs to check that the feasibility constraint (8.37) is satisfied. To aid in this process the constraints can be represented graphically, as shown in Figure 8.27. The steps for using this figure to design a UMZV$_C$ shaper are:

1. Use the system parameters τ_a, τ_b, and T to identify a point in Figure 8.27.
2. If the point is in the black region labeled "No Sol'n", then the parameters violate the constraint (8.37) and no solution is possible.
3. If the point is a feasible solution, then the figure indicates the minimum values of n and m. Regions of constant n and m are separated by solid lines and are shaded white and grey in an alternating pattern.
4. Substitute the values of n and m, as well as the other system parameters into (8.34) and (8.35) to complete the command design.

8.2.6 UMZV$_C$ Commands: Effect of Pulse Duration

Changing the pulse duration, t_p also effects the functionality of a UMZV$_C$ shaper. Recall from Figure 8.17 that t_p defined the duration of the baseline pulse command. In earlier discussions, commands were categorized as either *short*, *interference*, or *long* depending on the value of t_p. The same analysis can be applied here. The UMZV$_C$ command is designed to work for long commands because its derivation was based on the long command graphically represented in Figure 8.23. As a result, assumption (8.25) must hold.

To test the limits of the pulse duration, several simulations and experiments were conducted. The vibration induced by UMZV$_C$ commands was measured for various pulse

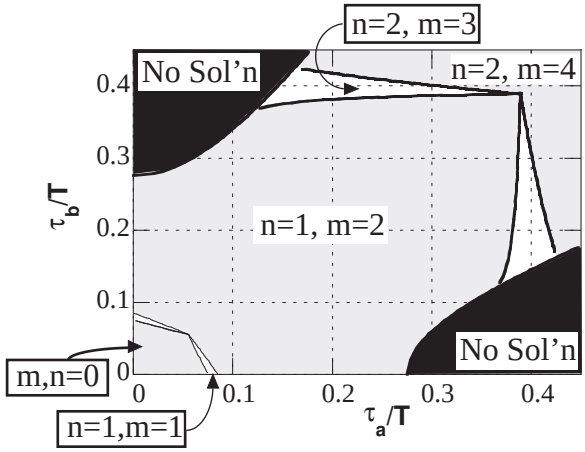


Figure 8.27: Constraints on τ_a/T and τ_b/T for a UMZV$_C$ Shaped Command.

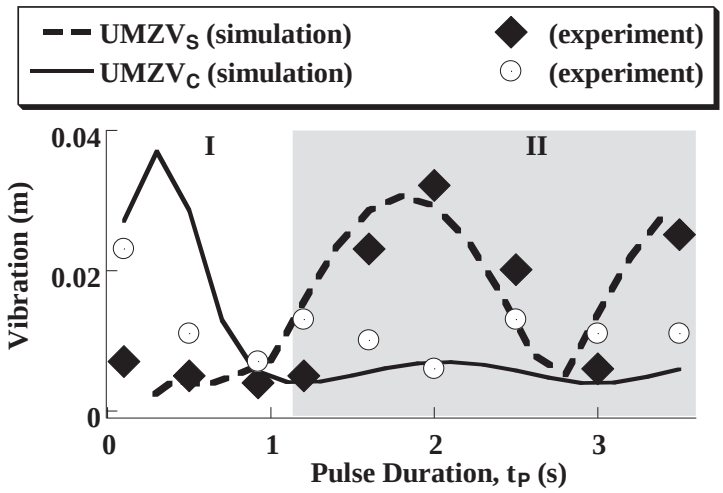


Figure 8.28: UMZV$_C$ Shaped and Unshaped Residual Vibration for Various Pulse Durations $[\tau_b = 0.065(s)]$.

durations. The results are shown in Figure 8.28, along with the UMZV vibration data from Figure 8.24 for comparison. The graph is divided into two regions. Region II contains long commands that satisfy (8.25), and region I contains all other commands. The vibration resulting from UMZV$_C$ commands in region II is much closer to zero than when standard UMZV commands are used. So, for long commands the UMZV$_C$ shaper is a substantial improvement over the standard UMZV shaper. However, in Region I UMZV$_C$ shaper performs poorly because the commands are short or interference. Both of these observations support the theoretical arguments above.

A second set of simulations and experiments were conducted in which both the pulse

duration and braking time constant were varied. The results are shown in Figure 8.29(a). Notice that for most of these values, the vibration is nearly zero. Again this is a substantial improvement over the standard UMZV data shown in Figure 8.25. To aid in the comparison between the UMZV and UMZV_C results, the average vibration across τ_b at each value of t_p was measured. The results are shown in Figure 8.29(b). This data was formed using the data previously shown in Figures 8.29(a) and 8.25. The lines represent the theoretical data and the symbols represent the experimental data. The figure shows that the UMZV_C commands have better performance than the UMZV commands if the pulse time is sufficiently long. However, for short commands the UMZV_C shaper has degraded performance and the standard UMZV shaper is superior. This figure supports the theoretical ideas developed earlier as well as the conclusions drawn previously from Figure 8.28. Note that since this plot shows data averaged for several shapers, and the shaper duration changes for these shapers, the exact boundary between *short*, *interference*, and *long* commands cannot be defined.

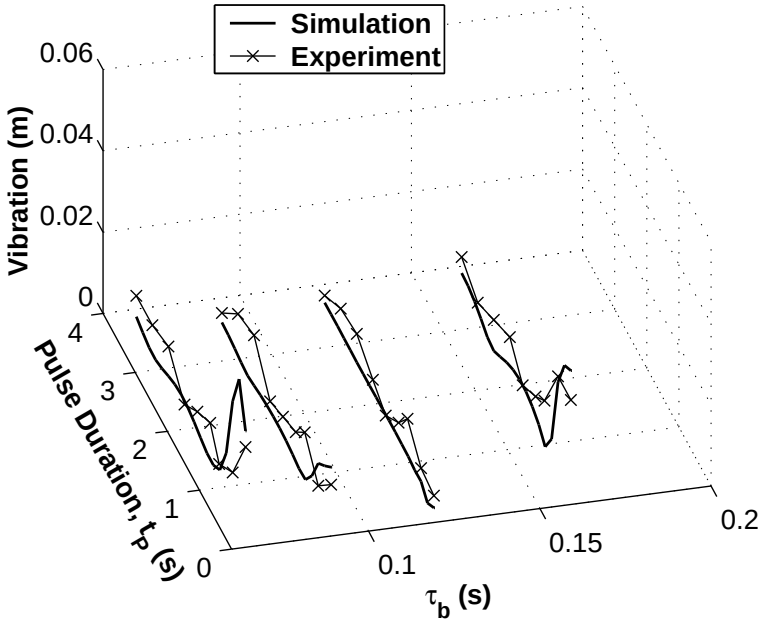
8.2.7 Conclusions

Several key ideas can be extracted from the nonlinear-braking study:

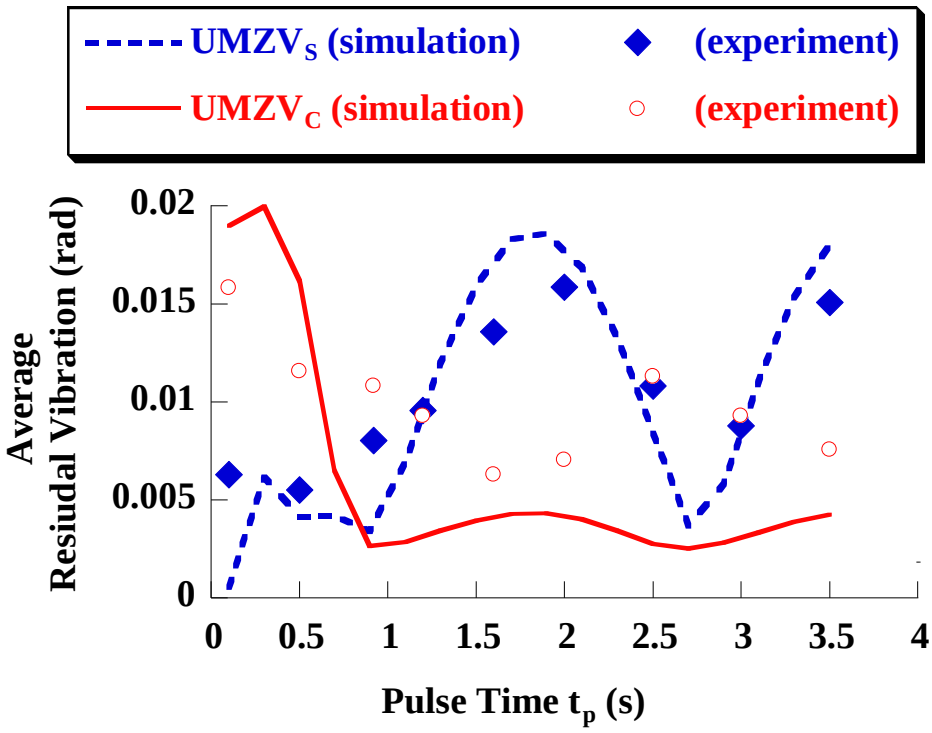
- The zero-vibration, vector solution is a valid means for generating input-shaped commands. The success of this technique has been proven in simulation and with experiments.
- Some shapers are more adversely affected by a nonlinearity than others. In this case, the UMZV shaper was more affected by the nonlinear braking than the ZV shaper.
- The command duration affects the performance of input shapers in nonlinear systems.
- For some combinations of system parameters a UMZV_C command may not exist.

8.3 Vector Approach for Digital Systems

In many cases, a continuous solution cannot be used. The nonlinear model of the drive may be too complicated, making it impossible to take the Laplace transform and solve the shaper

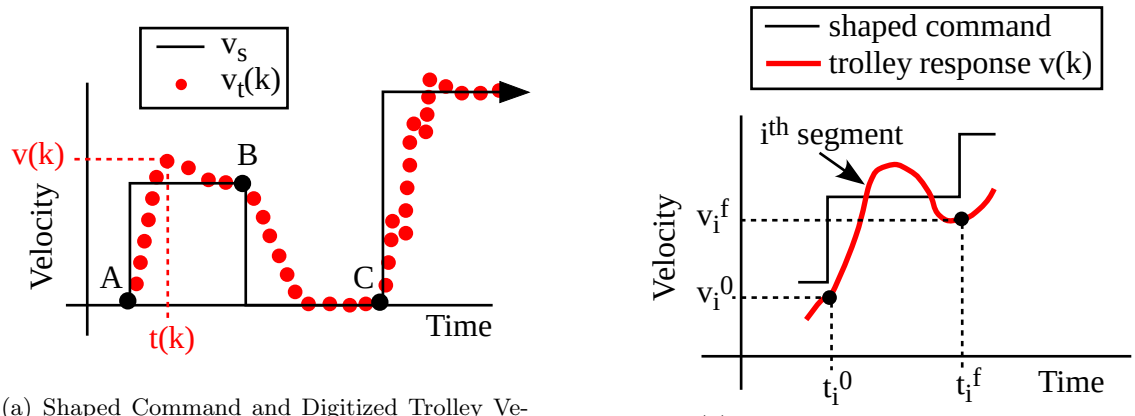


(a) Vibration of UMZV$_C$ Commands for Various t_p and τ_b



(b) Vibration of UMZV$_C$ Commands for Various t_p Averaged Across τ_b

Figure 8.29: Vibration of UMZV$_C$ Commands for Various t_p and τ_b.



(a) Shaped Command and Digitized Trolley Velocity

(b) Single Trolley Velocity Segment

Figure 8.30: Digitizing the Trolley Velocity.

design equations. In other cases, there may be no analytic model of the drive nonlinearity, only measurements of its behavior. In these cases, a digital algorithm is preferred. This section will develop such an algorithm and evaluate its performance through simulations and experiments.

8.3.1 Digitizing the Vector Solution

To digitize the vector based solution presented earlier, the $\vec{v}_i$ vectors must be derived for a discrete velocity signal. Figure 8.30(a) shows an example shaped command and the corresponding measured trolley velocity. However, the trolley velocity is reported as a series of sampled points, as shown in the figure. Each point is indexed by k, which has integer values from 0 to n. The time of each point is given by $t(k)$ and the value by $v(k)$. As was the case for continuous systems, the analysis begins by breaking up the trolley velocity into segments corresponding to the velocity transitions in the shaped command. In the figure, points A-C mark the locations where the command is divided.

Figure 8.30(b) shows a detailed sketch of an arbitrary command segment. Note that the trolley velocity is drawn as a line for convenience, but it is actually a series of sampled points. Using the figure as a guide, the following variables are defined:

i: Segment index.

k_i^0: The sample number of the first point in the i^{th} segment.

$\mathbf{t_i^0 = t(k_i^0)}$**:** The time of the first point in the i^{th} segment.

$\mathbf{v_i^0 = v(k_i^0)}$**:** The value of the first velocity point in the i^{th} segment.

$\mathbf{k_i^f}$**:** The sample number of the final point in the i^{th} segment.

$\mathbf{t_i^f = t(k_i^f)}$**:** The time of the final point in the i^{th} segment.

$\mathbf{v_i^f = v(k_i^f)}$**:** The value of the final velocity point in the i^{th} segment.

Note that the final time and velocity of the i^{th} segment are equivalent to the first time and velocity of the $i+1$ segment.

The next step is to form the g_i segments. Recall that this is achieved by shifting each segment in time and space so the first velocity point occurs at a time of zero and has value of zero. Then, the last value in the segment is extended for an infinite amount of time. Figure 8.31 shows the g_i construction for the segment previously shown in 8.30(b). As a result of the time and velocity shifting operations, a new set of variables are defined that are local each g_i segment. These variables can be seen in Figure 8.31:

$\mathbf{m_i}$**:** Local sample index for the i^{th} segment.

$\tau(\mathbf{m_i})$**:** The shifted time values for the i^{th} segment. These values are shift to begin at zero.

$\mathbf{g_i(m_i)}$**:** The shifted velocity values for the i^{th} segment. These values are shifted to begin at zero.

$\mathbf{m_i^f}$**:** The local sample number that corresponds to the last point in the i^{th} segment.

$\tau_\mathbf{i}^\mathbf{f} = \tau_\mathbf{i}(\mathbf{m_i^f})$**:** The τ-time that corresponds to the last point in the i^{th} segment.

$\mathbf{y_i = g_i(m_i^f)}$**:** The g_i-velocity that corresponds to the last point in the i^{th} segment.

Comparing Figures 8.30(b) and 8.31 the following relationships arise:

$$m_i^f = k_i^f - k_i^0$$

$$\tau_i^f = t_i^f - t_i^0 = \tau_i(m_i^f)$$

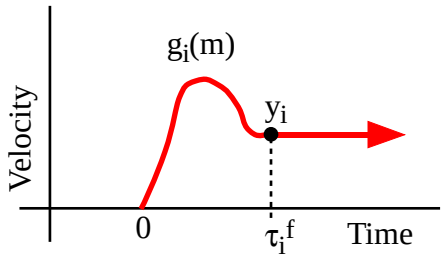
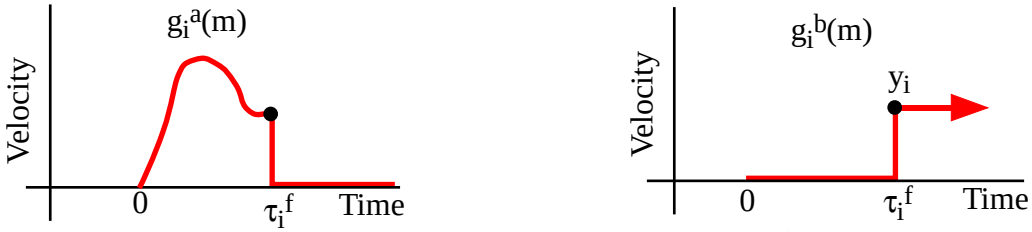


Figure 8.31: Original $g_i(k)$ Segment



(a) $g_i^a(k)$ Segment

(b) $g_i^b(k)$ Segment

Figure 8.32: Analysis of $g_i(t)$ segment.

$$y_i = v_i^f - v_i^0$$

$$\tau_i(m) = t(k_i^0 + m) - t_i^0$$

$$g_i(m) = \begin{cases} v(k_i^0 + m) - v_i^0, & \forall\ m \leq m_i^f \\ y_i, & \text{otherwise} \end{cases}$$

Now that the g_i are defined, the next step is to calculate $G_i(j\omega_n)$. For continuous systems this amounted to taking the Laplace transform of g_i, setting $s = j\omega_n$, and then evaluating. However, this process is equivalent to finding the complex Fourier transform of g_i at ω_n. Fourier transforms also have the advantage that they are easy to approximate for a digital signal.

To find the Fourier transform of g_i, the signal is broken up into the two pieces shown in Figures 8.32(a) and 8.32(b). The new segment components are denoted g_i^a and g_i^b respectively. The g_i^a component is equivalent to g_i for $\tau_i \leq \tau_i^f$ and is zero afterward. The g_i^b component is zero for $\tau_i \leq \tau_i^f$ and equivalent to g_i afterward. Therefore, the g_i segment is equivalent to the sum of these two profiles:

$$g_i = g_i^a + g_i^b \tag{8.38}$$

The Fourier transform of g_i is equal to the sum of the Fourier transform of each sub-component:

$$G_i(\omega) = G_i^a(\omega) + G_i^b(\omega) \tag{8.39}$$

The problem is now reduced to finding the Fourier transforms of g_i^a and g_i^b.

The Fourier transform of a signal sampled for an infinite length of time can be approximated using the Semi-Discrete Fourier Transform (SDFT)[1] [101]. Applying the SDFT to g_i^a yields:

$$G_i^a(\omega_n) = \Delta \sum_{m=0}^{m_i^f} e^{j\omega_n \tau_i(m)} g_i^a(m) \tag{8.40}$$

where Δ is the time between two samples. Note that the Discrete Fourier Transform (DFT) could have been used instead. However, the SDFT provides a better estimate of the Fourier transform than the DFT. This is because the DFT only exists at discrete values of ω, which may not include the natural frequency of the crane. The SDFT is continuous for all frequencies between zero and the Nyquist frequency, so the SDFT can be evaluated exactly at the crane's natural frequency.

The Fourier transform of the g_i^b segment can be determined exactly. Notice that g_i^b is nothing more than a delayed step, which has a well-defined Fourier transform:

$$G_i^b(\omega_n) = \frac{y_i}{j\omega_n} e^{-j\tau_i^f \omega_n} \tag{8.41}$$

Substituting (8.40) and (8.41) into (8.39) yields the Fourier transform of the whole g_i segment at ω_n:

$$G_i(\omega_n) = \Delta \sum_{m=0}^{m_i^f} e^{j\omega_n \tau_i(m)} g_i^a(m) + \frac{y_i}{j\omega_n} e^{-j\tau_i^f \omega_n} \tag{8.42}$$

Once $G_i(\omega_n)$ is known, the remainder of the vector analysis is the same as in the continuous case. The $G_i(\omega_n)$ values are substituted into (8.12) to find the v_i vector values. Then, (8.10) and (8.11) are used to find the amplitude and phase of the steady-state response. As a side note, the notation $G_i(\omega_n)$ represents the Fourier transform of g_i evaluated at ω_n.

[1] It is assumed that the sampling frequency is larger than the Nyquist frequency. This is a realistic assumption since a crane's period is often on the order of seconds and the sampling period would normally be at least an order of magnitude higher.

However, the $G_i(j\omega_n)$ notation in (8.12) represents the Laplace transform g_i evaluated at $s = j\omega_n$. As mentioned earlier these two are equivalent, so they are used interchangeably.

8.3.2 Digital Analysis Tool

In order to develop a useful solution technique we will create a visual analysis tool to aid in the design process. The tool consists of two plots; a vector diagram plot and a frequency response plot. The template shaping strategy developed earlier finds a nonlinear shaper that mimics the performance of a corresponding linear shaper. The tools developed in this section are a visual aid for comparing the performance of these two shapers.

The first visual aid is a vector plot that compares the response vectors of the template shaper and the nonlinear shaper. Recall that the relationship between these two sets of vectors consists of a global and local scaling and a global and local rotation. The vector plot developed here is designed to only show the local shift since this the represents the degrading effects of the nonlinearity on the shaper performance. The global shift is not shown since it is present in linear systems and does not degrade shaper performance. Recall that the global scaling and shifting terms are contained in the Q and R constants from (8.20). Therefore, their effects on the $\vec{v}_i$ vectors can be removed by dividing all the vector magnitudes by Q and subtracting R from all the vector angles.

Figure 8.33 gives an example of what this vector plot would look like. The solid arrows represent the template shaper, and the dashed arrows represent the nonlinear shaper. The horizontal axis is the vector angle, and the vertical axis is the vector amplitude. Notice that the vector plot is an cartesian plot, rather than the traditional polar plot. This is because it is easier to measure and compare the precise angle and amplitude values on a cartesian plot.

The vector plot for the template shaper is formed using the same rules developed in the literature [85]. The only difference is that the vector angles are "unwrapped" onto the horizontal axis. Therefore, the amplitude and angle values from Figure 8.33 become:

$$\gamma_i = \omega_n \tilde{t}_i \tag{8.43}$$

$$C_i = \tilde{A}_i \tag{8.44}$$

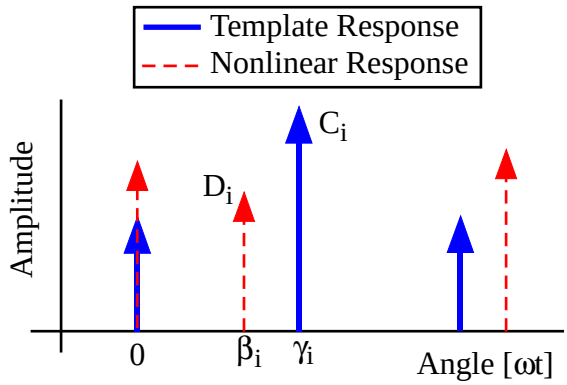


Figure 8.33: Vector Diagram Tool for Nonlinear Systems.

where $\tilde{A}_i$ and $\tilde{t}_i$ are the amplitudes and times of the template shaper.

The vectors representing the nonlinear shaper, $\vec{v}_i$, were previously derived in (8.12). However, recall that the vectors in a vector diagram differ in that the $\omega_n t_i$ term for the angle is positive instead of negative and the vector amplitudes are allowed to be negative. These must be removed to perform a meaningful comparison with the standard vector diagram vectors, $\vec{x}_i$. In addition, the global scaling and rotation of the $\vec{v}_i$ vectors are also removed, as mentioned above. Making these adjustments to the $\vec{v}_i$ vectors from (8.12) yields the amplitudes and times for the nonlinear vectors in Figure 8.33:

$$\beta_i = \omega_n t_i - \angle[G_i(\omega_n)] + m_i\pi + (\angle G_1(\omega_n) - m_1\pi) \tag{8.45}$$

$$D_i = \frac{p_i|G_i(\omega_n)|}{Q} \tag{8.46}$$

where,

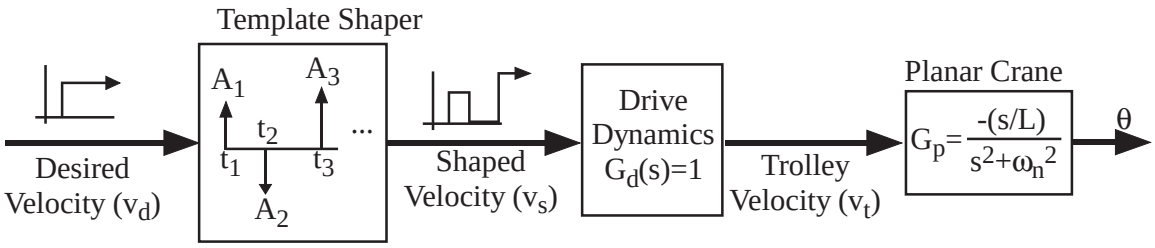
$$Q = \sum p_i|G_i(\omega_n)|, \qquad p_i = \text{sign}(\tilde{A}_i), \qquad m_i = \begin{cases} 0, & \text{if} \tilde{A}_i \geq 0 \\ 1, & \text{otherwise} \end{cases}$$

The physical significance of each term is:

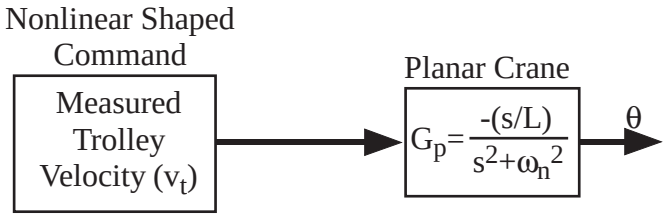
$G_i(\omega_n)$**:** The Fourier transform of the g_i segments given by (8.42).

t_i**:** The step times of the nonlinear shaped command.

Q**:** Scales the amplitudes so they add to one. This is related to the Q term from Section 8.1. Dividing it out can also be viewed as removing the global gain from the nonlinear drive dynamics.



(a) Block Diagram for Forming Response to Template Shaper.



(b) Block Diagram for Forming Response to Nonlinear Shaped Command.

Figure 8.34: Block Diagrams for Forming the Template Shaper Response and Nonlinear Response.

p_i**:** Allows the nonlinear vectors to have negative amplitude, similar to the template vectors.

m_i**:** Compensates for negative amplitude vectors in the phase.

$\angle G_1(\omega_n) - m_1\pi$**:** Shifts the angle of the nonlinear vectors so the first vector always has an angle of zero. This can also be viewed as extracting the global phase shift due to the dynamics of the nonlinear drive.

$\tilde{A}_i$**:** The amplitudes from the template shaper.

The second visual aid is a frequency response plot, similar to the one previously shown in Figure 8.16. The template shaper is usually chosen based on its frequency characteristics such as robustness, or zero vibration at specific frequencies. Therefore, it will be useful to plot the frequency response of the nonlinear shaper and compare it to the frequency response of the template shaper. For the template shaper, this means finding the shaped response of the linear crane system, shown in Figure 8.34(a), at different frequencies. Notice that the drive system is assumed to achieve perfect tracking and has a transfer function of 1. The vibration amplitude for various frequencies can be determined using either a linear simulation program, or the closed-form solutions given previously. Figure 8.34(b) shows

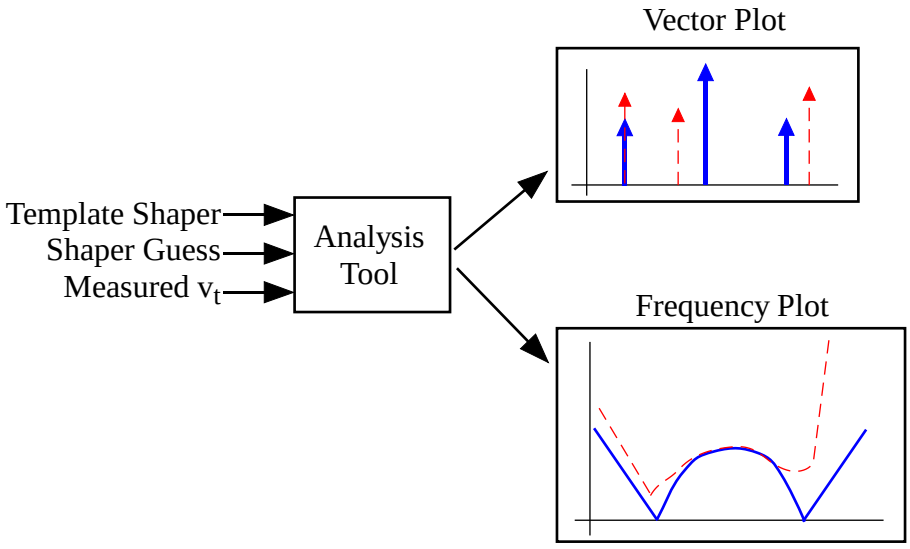
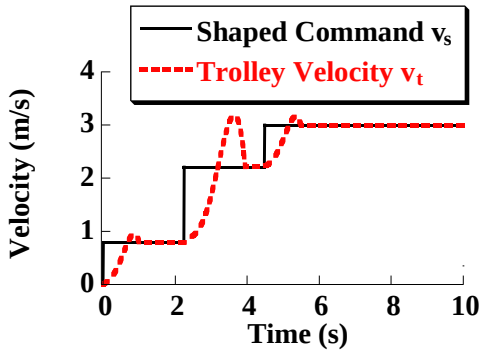


Figure 8.35: Analysis Tool for Nonlinear Systems.

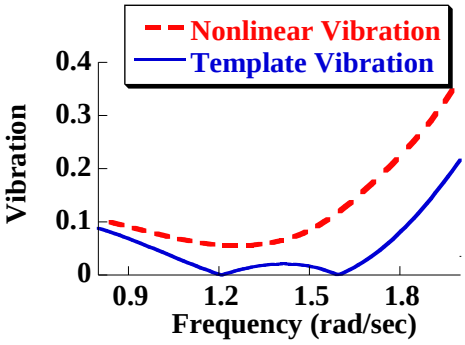
the block diagram used to find the response of the nonlinear system. Because the trolley velocity is already known, the payload response can be determined with either a linear simulation program, or the closed-form solutions given earlier.

An overview of these analysis tools is shown in Figure 8.35. The required inputs are the template shaper, the nonlinear shaper, and the measured trolley velocity. These are sent to a program, labeled "Analysis Tool" in the figure. This program calculates the vector values of the shaper template and nonlinear shaped command using equations (8.43)-(8.46). It also calculates the frequency response of the template shaper and nonlinear shaped command, using the methods described above. The output is the vector diagram and frequency response, as described above.

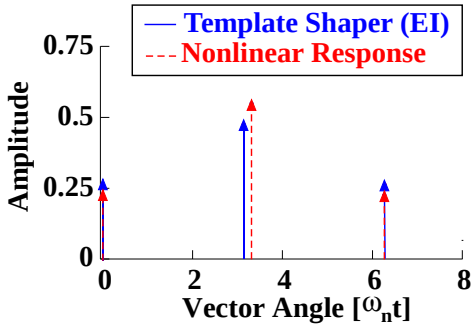
To give an example of how this analysis tool is used, consider the nonlinear trolley response shown in Figure 8.36(a) with the red-dotted line. The shaped command that was sent to the drive is an EI-shaped step, shown with the solid line. The details of the nonlinear drive simulation are described in Section 8.4. In this case, the template shaper was an EI shaper. The same shaper was used as a "first-guess" for the nonlinear shaper. Applying the analysis tool to the data, as shown in Figure 8.35, yields Figures 8.36(b) and 8.36(c). The frequency plot clearly shows that the nonlinearity has degraded the intended performance of the EI shaper. The vector representation gives more insight into why the



(a) Desired and Actual Trolley Command.



(b) Frequency Response.



(c) Vector Diagram.

Figure 8.36: Example of using the Nonlinear Analysis Tool.

response is degraded. The more amplitude or phase difference there is between the template shaper and nonlinear vectors, the larger the discrepancy will be between the two frequency responses.

8.3.3 Digital Solution Techniques

Earlier it was shown how vector analysis of nonlinear systems can be used to find input shaped commands for continuous systems. The last two sections have demonstrated that a similar vector analysis can be applied to digital systems. Using these digital vector-analysis tools, solutions for input shaped commands can be formed, similar to continuous systems. Two solution techniques are developed. The first is a trial-and-error method that uses the vector diagram analysis tool from the last section. The second technique attempts to numerically tune the vectors. Both of these techniques can obviously be extended to

continuous systems by sampling the trolley velocity.

8.3.3.1 Trial-and-Error Vector Matching

The vector plot gives immediate insight into how the nonlinear shaper response can be improved. The objective is to shift the amplitude and angle of the nonlinear vectors to match the template vectors. This process is based on two simple ideas:

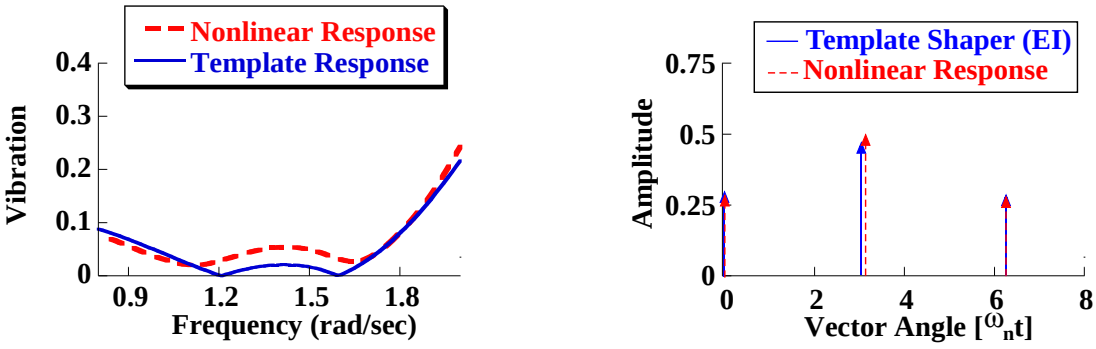
- The amplitude of any nonlinear vector can be increased (or decreased) by increasing (or decreasing) the amplitude of the corresponding impulse in the nonlinear shaper.
- The angle of any nonlinear vector can be increased (or decreased) by increasing (or decreasing) the time of the corresponding impulse of the nonlinear shaper.

For the majority of the nonlinear drive systems, these basic premises should hold. However, there are some nonlinear systems where they do not hold and these are outside the realm of this type of solution. In addition, if there is coupling between the $\vec{v}_i$ vectors the solution will be more difficult to find.

To demonstrate how to apply this process, consider the nonlinear command and response shown earlier in Figure 8.36. The nonlinear response vectors need to be shifted to match the amplitude and angle of the template shaper. The amplitude shift is discussed first, then the phase shift.

According to Figure 8.36(c), the second dashed vector needs to be decreased in amplitude, and the first and third red vectors need to be increased in amplitude. This suggests how to modify the amplitude of the shaper impulses. However, the new amplitudes must be chosen so they still sum to one. The amount to decrease, or increase each vector can be estimated by the ratio of the template and nonlinear vectors. For example, if the 2^{nd} template vector is 5% smaller than the 2^{nd} nonlinear vector, then the 2^{nd} impulse amplitude of the nonlinear shaper should be decreased by 5%.

A similar method can be used to find the impulse times. Figure 8.36(c) shows that the second nonlinear vector is leading the second template vector. Therefore, the time of the second impulse should be decreased. As with the amplitudes, the amount by which to



(a) Frequency Response.

(b) Vector Diagram.

Figure 8.37: Nonlinear Shaper Performance after 1 iteration.

decrease, or increase the time should be governed by the vector plot. For example, suppose the 2^{nd} template vector has a angle that is 0.1 radians larger than the angle of the 2^{nd} nonlinear vector. Because the angle is given by $\omega_n t_i$, this suggests that the time of the second vector should be decreased by $\frac{0.1}{\omega_n}$.

The empirical method outlined above was applied to the results from Figure 8.36. The system was then re-simulated with the new nonlinear compensating shaper. Figure 8.37 shows the results. The frequency response of the nonlinear system, shown in Figure 8.37(a), is much closer to the EI template. In addition, the vector plot in Figure 8.37(b) shows that the nonlinear vectors are closer to the template vectors. This process could be repeated for as many iterations as necessary to improve the nonlinear shaper performance.

8.3.3.2 Automated Vector Matching

The trial-and-error process outlined above can be easily implemented by a computer using basic numerical techniques. This is useful for streamlining calculations and running multiple iterations. The idea is to develop a simple algorithm for estimating the next guess for the nonlinear shaper parameters. As was the case for the trial-and-error process, the next guess is formed by trying to set the vector angles and amplitudes of the nonlinear shaper equal to the template shaper.

The next guess for the amplitude values on the nonlinear compensating shaper impulses can be found by solving a linear algebra problem. During this process, the current nonlinear

shaper amplitudes are given by A_i^0, while the amplitudes for the next guess are given by A_i^1. Likewise, the current vector diagram amplitudes are given by D_i^0, and the estimate of the vector diagram amplitudes for the next iteration are given by D_i^1. The template shaper amplitudes are C_i and they remain unchanged during the processes. The amplitudes must be constrained to sum to 1, yielding:

$$\sum A_i = 1 \tag{8.47}$$

The key assumption of this prediction algorithm is that the vector amplitude, D_i, is proportional to the shaper amplitude, A_i:

$$D_i = A_i \cdot r_i \tag{8.48}$$

where r_i is the constant of proportionality. This is exactly true for linear systems and should be a good approximation for small changes in A_i for the nonlinear system. The current amplitudes can be used to find the constant of proportionality:

$$r_i = \frac{D_i^0}{A_i^0} \tag{8.49}$$

Next, the D_i^1 amplitudes are set equal to C_i for the next iteration. Equations (8.48) and (8.49) can be used to relate D_i^1 to A_i^1. Recall that when the D_i^1 vectors are formed, they are normalized by the constant Q so that they sum to 1. Including this extra factor yields:

$$C_i = D_i^1 = \frac{A_i^1 r_i}{Q} = \frac{A_i^1 D_i^0}{Q A_i^0} \tag{8.50}$$

Equations (8.47) and (8.50) can be rearranged into a matrix equation:

$$\left[\begin{array}{ccc|c} & & & \vdots \\ & \mathbf{I}_{n,n} & & -\frac{C_i A_i^0}{D_i^0} \\ & & & \vdots \\ \hline 1 & \cdots & 1 & 0 \end{array}\right] \left[\begin{array}{c} \vdots \\ A_i^1 \\ \vdots \\ \hline Q \end{array}\right] = \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline 1 \end{array}\right] \tag{8.51}$$

where $\mathbf{I}_{n,n}$ is an n by n identity matrix and n is the number of shaper impulses. Solving this matrix equation gives the A_i^1 amplitudes for the next iteration.

The next guess for the nonlinear shaper times is also found by solving a series of algebraic equations. The main assumption here is that the phase shift due to the nonlinearity is constant:

$$\beta_i = \omega_n t_i + s_i \tag{8.52}$$

where s_i is a constant that represents the phase shift between $\omega_n t_i$ and β_i. Recall from (8.45) that the shift between $\omega_n t_i$ and β_i is not constant due to the $G_i(\omega_n)$ term that can vary with A_i or t_i. However, for small changes in t_i it will be assumed that the $\omega_n t_i$ term dominates.

Assume that the β_i values for the current iteration are given by β_i^0 and the values for the next iteration are given by β_i^1. Likewise the current nonlinear shaper times are given by t_i^0 and the shaper times for the next iteration are t_i^1. The β_i^0 and t_i^0 values from the previous iteration can be used to solve for the s_i phase shift:

$$s_i = \beta_i^0 - \omega_n t_i^0 \tag{8.53}$$

The next iteration strives to make the β_i^1 angles from the nonlinear shaper equal to the γ_i angles from the template shaper. Using this in conjunction with (8.52) and (8.53) yields:

$$t_i^1 = t_i^0 + \frac{\gamma_i - \beta_i^0}{\omega_n} \tag{8.54}$$

Solving (8.54) gives the t_i^1 times for the next iteration.

The digital, nonlinear shaper algorithm is now complete. The amplitudes and times for the next iteration of the nonlinear shaper are easily found using (8.51) and(8.54). It should be noted that this digital solution is based on the same ideas as the solution for continuous systems presented earlier. The amplitude and phase of the normalized nonlinear shaper vectors are set equal to the template shaper vectors.

In some cases the iteration process may be divergent or "jittery". To fix this problem, the times and amplitudes of the shaper can be changed by a fraction of the full amount recommended by the algorithm. More precisely, suppose that equations (8.51) and(8.54) determine that the shaper amplitudes and times need to be changed by ΔA_i and Δt_i respectively. Rather than change by these full amounts, the amplitudes and times can be

changed by $\kappa_1 \Delta A_i$ and $\kappa_2 \Delta t_i$ respectively, where $0 < \kappa_1, \kappa_2 \leq 1$. This reduces the step size of each iteration which usually makes the process more stable. Note that since the ΔA_i values are uniformly scaled by κ_1, it will not violate the summation constraint on the amplitudes (8.47).

The next two sections use this automated algorithm to solve for shapers for nonlinear systems. In the first section several simulations show how the algorithm behaves across a large range of drive-nonlinearities. In the second section the algorithm is applied to the experimental bridge crane setup. Both sections show that the algorithm not only finds nonlinear shapers that perform well, but also converges to the solution in a very small number of iterations.

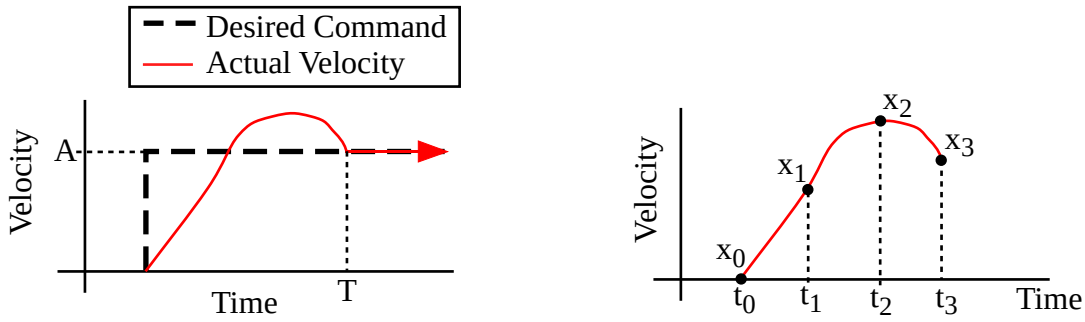
8.4 Application: Simulations of a Randomly Generated Drive nonlinearity

Several simulations were conducted to test the functionality of the nonlinear shaper algorithm. The first stage of this study developed a way to simulate nonlinear drive dynamics. In the second stage, the nonlinear shaper algorithm was used to generate shapers for the nonlinear drive. The simulations were run multiple times, using a different nonlinearity for each set of trials to show the robustness of the algorithm.

8.4.1 Simulating the Nonlinear Drive Dynamics with Polynomials

Before the nonlinear shaper simulations could be performed, a method for simulating a drive nonlinearity was developed. The simulation accomplishes the following:

- The trolley velocity profile varies with the step size.
- The trolley velocity settles in a finite time, T. After time T the command equals the desired velocity A.
- The simulation is controlled by a small number of parameters.
- The parameters form a simple, intuitive relationship with the trolley velocity response.



(a) Desired Command and Actual Velocity.

(b) Defining Points for a Polynomial Fit.

Figure 8.38: Forming a Randomly-Generated Drive Nonlinearity.

Figure 8.38(a) shows an example of what the input/output profiles for such a process would look like. Given a step command of amplitude A, shown as the dashed line, the simulation needs to output the trolley velocity, shown as the solid line.

The simulation uses polynomials to control every aspect of the output trolley velocity profile. The character of the nonlinearity can then be tuned by adjusting the coefficients of the polynomials. The settling time, T, was related to the step size, A, using the polynomial:

$$T = c_1 A^p + c_2 A^{p-1} + c_p A + c_{p+1} \tag{8.55}$$

where the polynomial is of order p+1.

Once the settling time is chosen, the velocity profile is already defined for all $t \geq T$. What remains is to form the command for $0 < t < T$, as shown in Figure 8.38(b). The command was discretized into n+1 equally spaced points, as shown in the figure. The times of each point are given by:

$$t_i = \left(\frac{T}{n+1}\right) i, \qquad \forall \;\; i = 0 \ldots n+1 \tag{8.56}$$

Note that $t_0 = 0$ and $t_{n+1} = T$.

Next, polynomials were used to relate the x_i values at each point to the step size, A:

$$\begin{aligned} x_1 &= b_{1,1} A^m + b_{1,2} A^{m-1} + \ldots + b_{1,m} A \\ x_2 &= b_{2,1} A^m + b_{2,2} A^{m-1} + \ldots + b_{2,m} A \\ &\vdots \\ x_n &= b_{n,1} A^m + b_{n,2} A^{m-1} + \ldots + b_{n,m} A \end{aligned} \tag{8.57}$$

where each of these polynomials is of order m+1. Notice that there is no constant term in these equations because as $A \to 0$ it is assumed that $x_i \to 0$. The entire profile can be formed by fitting a n+2 degree polynomial to the n+2 points $x_0 \ldots x_{n+1}$. Analytically this can be achieved using Lagrange polynomials:

$$x(t) = \sum_{i=0}^{n+1} x_i \prod_{\substack{j=0 \\ j \neq i}}^{n+1} \frac{(t - t_j)}{(t_i - t_j)} \tag{8.58}$$

The response to multi-step commands are constructed by recursive application of these equations. If the trolley velocity has not settled by the time of the next step, then the next step size, A_{i+1}, is taken to be the difference between the last trolley velocity point and the new commanded velocity.

The nonlinear drive simulation, described above, has several properties that are worth mentioning:

- The process is time independent
- The process only depends on the *change* in the commanded velocity, not the absolute commanded velocity values.
- For this process to be linear there must exist some $z(t)$ such that $x(t, A) = Az(t)$. The only way for this to occur is if $T = c_{p+1}$ and $x_i = b_{k,m}A$. Alternatively this condition can be stated as $p = 0$ and $m = 1$. This can be confirmed by substituting these relations into (8.58).

To summarize, the nonlinear drive simulation is controlled by the following parameters:

$$[c_1 \ \ldots \ c_p] = \text{ The coefficients that define the settling time T.}$$

$$\begin{bmatrix} b_{1,1} & \cdots & b_{1,m} \\ \vdots & \ddots & \vdots \\ b_{n,1} & \cdots & b_{n,m} \end{bmatrix} = \text{ The coefficients that define the } x_i \text{ of the nonlinear response curve.}$$

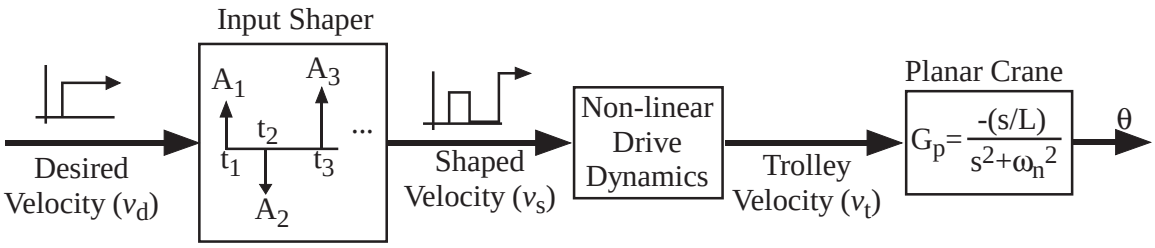


Figure 8.39: Nonlinear Drive System Block Diagram.

8.4.2 Simulation Setup

Figure 8.39 shows a block diagram of the system used for the simulation tests. The desired trolley velocity was set to $v_f = 0.5\ m/s$, and the suspension length was set to $L = 5\ m$. The "nonlinear drive dynamics" block contains the polynomial algorithm described above.

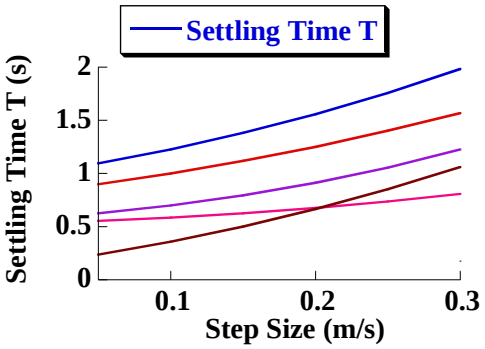
The nonlinear drive simulation is controlled by two sets of parameters, c_i and $b_{j,k}$. These parameters were chosen randomly by the computer to ensure a wide, unbiased variety of simulations. The c_i parameters were randomly chosen from the interval:

$$[0\ 0\ 0\ .01] \ldots [2\ 10\ 2\ 1]$$

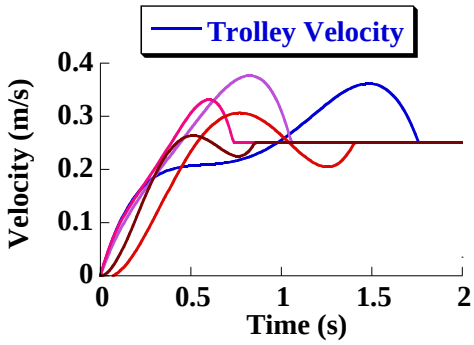
where the left array gives the lower bound of the c_i parameters, and the right array gives the upper bound. The $b_{j,k}$ parameters were randomly chosen from the interval:

$$\begin{bmatrix} 0.4 & -0.2 & 0.4 \\ -0.2 & -0.2 & 0.6 \\ 0 & -0.2 & 0.8 \end{bmatrix} \cdots \begin{bmatrix} 2 & 2 & 0.4 \\ 4 & 2 & 0.8 \\ 6 & 2 & 1 \end{bmatrix}$$

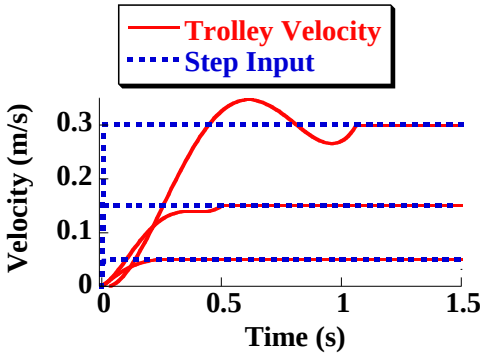
where the left matrix gives the lower bound of the $b_{j,k}$ parameters, and the right matrix gives the upper bound. Figure 8.40 shows the variety of commands that are generated by these values. The top-left plot, Figure 8.40(a), shows some of the possible settling times. Each curve was generated by randomly selecting c_i from the aforementioned range and evaluating the settling time, T, for each step size. The top-right plot, Figure 8.40(b), shows the variety of trolley velocity commands for the same step size. Each curve was generated by randomly selecting the $c_i, b_{j,k}$ from the aforementioned ranges and then evaluating the response to a 0.25 m/s step command. This clearly shows that the random selection of



(a) Settling Time vs. Step Amplitude



(b) Step Responses Using Randomly Chosen b and c values.



(c) Response to Various Step Sizes

Figure 8.40: Responses of Random Drive Nonlinearity Algorithm.

the parameters creates a wide variety of responses. The bottom plot, Figure 8.40(c), shows how the response can change for different step sizes. The c_i and $b_{j,k}$ parameters were held constant for these curves. The step size, shown as the dotted line, was varied yielding the plots shown. The response clearly varies with the step size in a nonlinear manner.

Figure 8.41 gives an overview of the simulation process. The first step is to choose all the initial parameters: the system parameters, the template shaper, and the nonlinear drive parameters c_i and $b_{j,k}$. The initial guess for the nonlinear shaper is the template shaper. All these values are sent to the nonlinear shaper solver. The solver performs a simulation of the system, it uses the shaper analysis tool to generate a vector diagram, and then it makes an initial guess for what the next shaper should be. After the fifth iteration the process was stopped and the performance of the nonlinear shaper was recorded. This process was

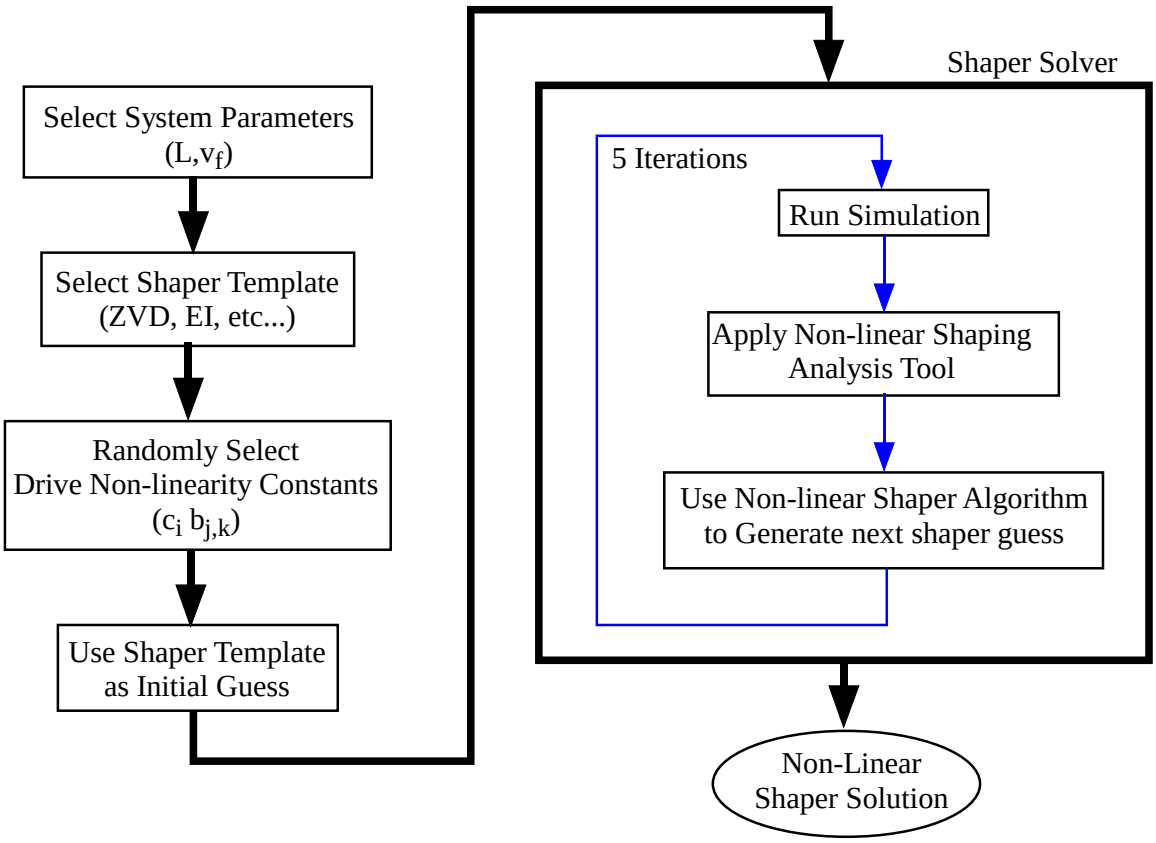
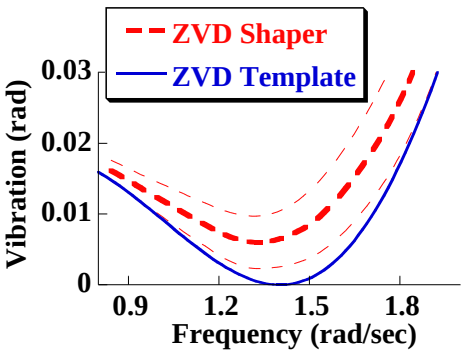


Figure 8.41: Simulation Overview

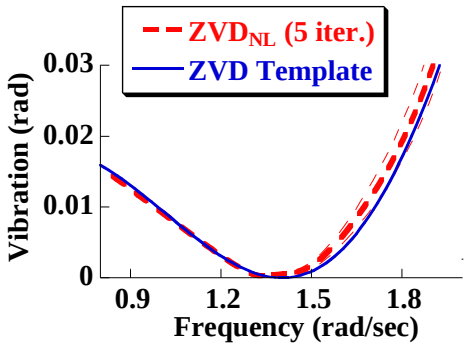
repeated 50 times using the same template shaper and system parameters, but randomly choosing a different set of nonlinear drive parameters each time. As a result, the shaper algorithm was tested using a large variety of different drive nonlinearities.

8.4.3 Simulation Results

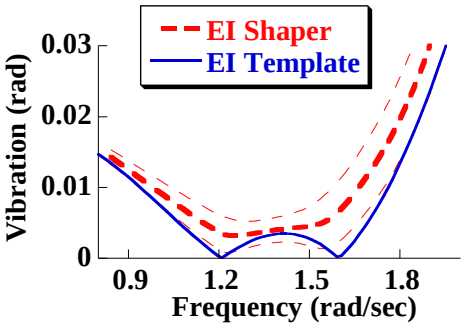
The simulation results are shown in Figure 8.42. All of the plots show the frequency response, as defined in Section 8.3.2. In the first set of 50 runs, a ZVD shaper was used as the template shaper. Figure 8.42(a) shows the response of standard ZVD shapers. The solid line is the desired, ZVD template response assuming the drive transfer function is unity (perfect tracking). Recall that for all 50 trials, the template stays exactly the same. The thick-dashed line is the vibration when the ZVD shaper is used on the nonlinear system. The vibration at each frequency is averaged across the 50 trials to yield each point in the thick-dashed line. The thin-dashed lines show the standard deviation. It is clear that the



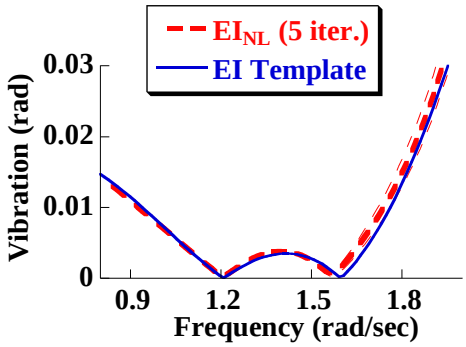
(a) Standard ZVD Shaper.



(b) Nonlinear ZVD Shaper.



(c) Standard EI Shaper.



(d) Nonlinear EI Shaper.

Figure 8.42: Comparison of Standard and Nonlinear Shapers after 5 iterations.

nonlinearity has degraded the response.

Figure 8.42(b) shows the frequency response of the nonlinear shaper based on a ZVD template shaper. The new shaper is labeled ZVD_{NL}. The solid line is the desired, ZVD template response that was shown in Figure 8.42(a). The thick-dashed line is the average response of the nonlinear shapers for the 50 trials. The thin-dashed lines also show the standard deviation of the 50 trials. Not only is the average much closer to the template, but the standard deviation is much smaller also.

In the second set of 50 trials, an EI shaper was used as the template. Figure 8.42(c) shows the response of a standard EI shaper. Note that by comparing Figures 8.42(a) and 8.42(c) it is apparent that the superior robustness of the EI shaper carries over into the nonlinear system. Figure 8.42(d) shows the response of the nonlinear shaper. Once again

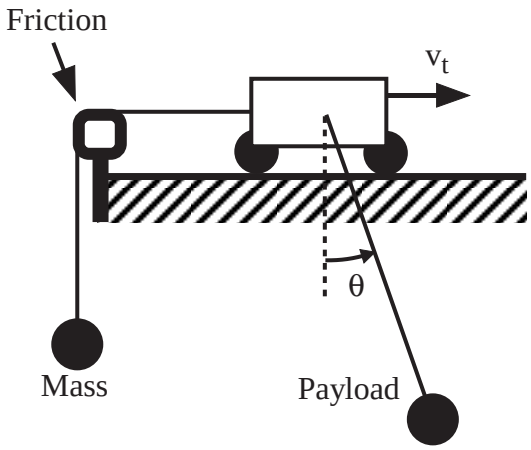
the nonlinear shaper shows excellent performance. The average response is closer to the desired template, and the standard deviation is reduced.

These simulation results clearly show the effectiveness of the nonlinear shaper algorithm for a wide range of scenarios. In the next section the algorithm is applied to the portable bridge crane.

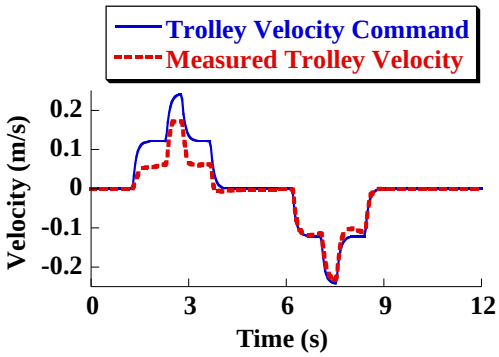
8.5 Application: Low Gain Bridge Crane Experiments

The portable bridge crane was used to test the digital nonlinear shaper algorithm. Figure 8.43(a) shows a sketch of the experimental setup. A string was tied to the trolley and draped over a metal bar with a mass tied to the other end. This added a constant gravity force as well as a Coulomb friction force to the system. To accentuate the nonlinear dynamics of the drive system, the PI gains of the motor drives were set to low values. Figure 8.43(b) shows a ZV shaped command and the trolley response. The command makes the system move forward, pause, than move backward. This command is common not only to cranes, but pick-and-place robots as well. Notice that in the forward direction there is significant deviation between the desired and actual trolley velocity. However, in the reverse direction there is much less tracking error. The significant difference between the forward and backward trolley response is evidence of the nonlinear behavior of the system. Figure 8.43(c) shows the measured payload response. The figure shows that there is significant vibration both during and after the ZV command. This shows that the performance of the ZV shaper is degraded by the nonlinear dynamics of the system.

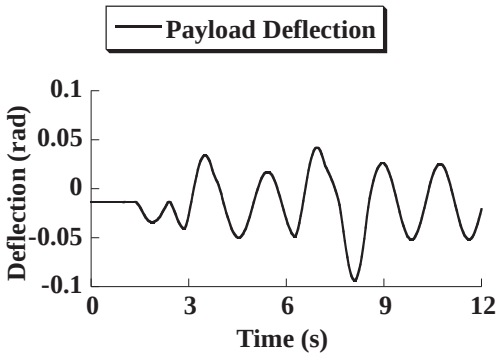
The digital nonlinear shaper algorithm was applied to the above system to improve the response. Figure 8.43(d) shows the resulting compensated command. Notice that the compensated command uses a larger first step for the positive velocity motion. This compensates for the offset in the trolley velocity. The negative velocity motion of the compensated command is very similar to the original ZV command. This is because the trolley tracks the negative portion of the command well and therefore little alteration is necessary. The response to this compensated command is shown in Figure 8.43(e). The response to the compensated command has significantly less vibration than the original ZV



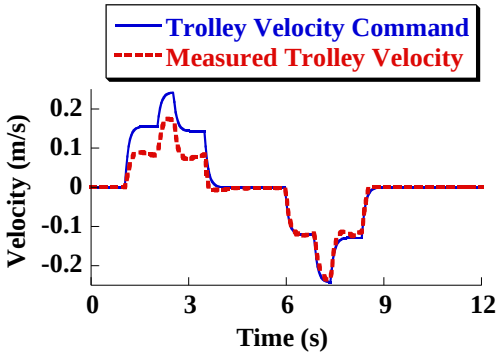
(a) Experimental Setup.



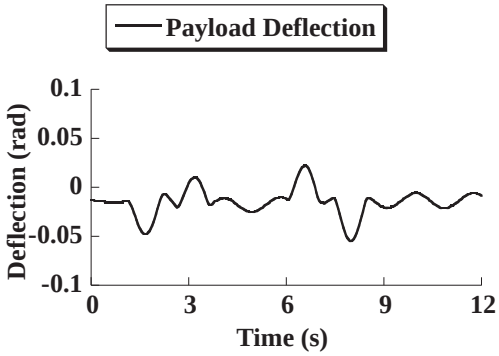
(b) Original ZV Shaped Command.



(c) Response to Original ZV Shaped Command.



(d) Compensated ZV Shaped Command.



(e) Response to Compensated ZV Shaped Command.

Figure 8.43: Bridge Crane Low-Gain Experiments: Setup and Experimental Data.

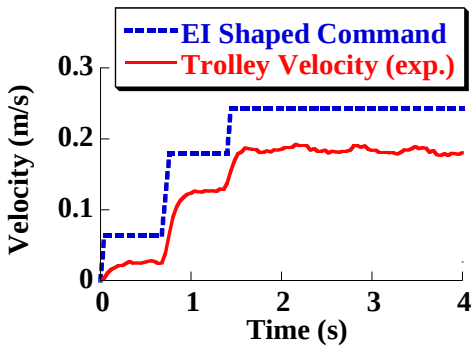


Figure 8.44: Bridge Crane Response to EI Command.

command. The residual vibration has been decreased by 80% and there is less transient vibration as well.

The previous set of experiments showed that the digital nonlinear shaper algorithm caused a significant reduction in the amount of residual vibration. The next set of experiments examine whether the nonlinear shaper algorithm can be used to create robust commands. Rather than use the full forward-pause-backward command sequence, only the ramp-up-to-speed segment of the command was tested. Figure 8.44 shows an example of the ramp-up-to-speed segment for a EI shaped command. The entire forward-pause-backward command was not used because the command contains three arbitrary times: the coasting time at positive velocity, the coasting time at negative velocity, and the length of time the system is stopped in the middle of the command. The choice of these three times effects the residual vibration of the command due to constructive/destructive interference of the vibrating payload. For example, the times could be chosen to yield a zero vibration response regardless of the performance of the input shaper. To avoid skewing the data from this effect, only the single ramp-up-to-speed segment was used. If the nonlinear shaping algorithm is shown to be robust when applied to this segment, it is assumed that the algorithm would show comparable robustness when applied to the other segments of the command.

Three different template shapers were used to form the compensated command: ZV, ZVD, and EI. Figure 8.45 shows the procedure used to find the nonlinear shaper solution. The process is similar to the one used in the previous section for the simulated system. The template shaper was used to form an initial guess for the nonlinear shaped command.

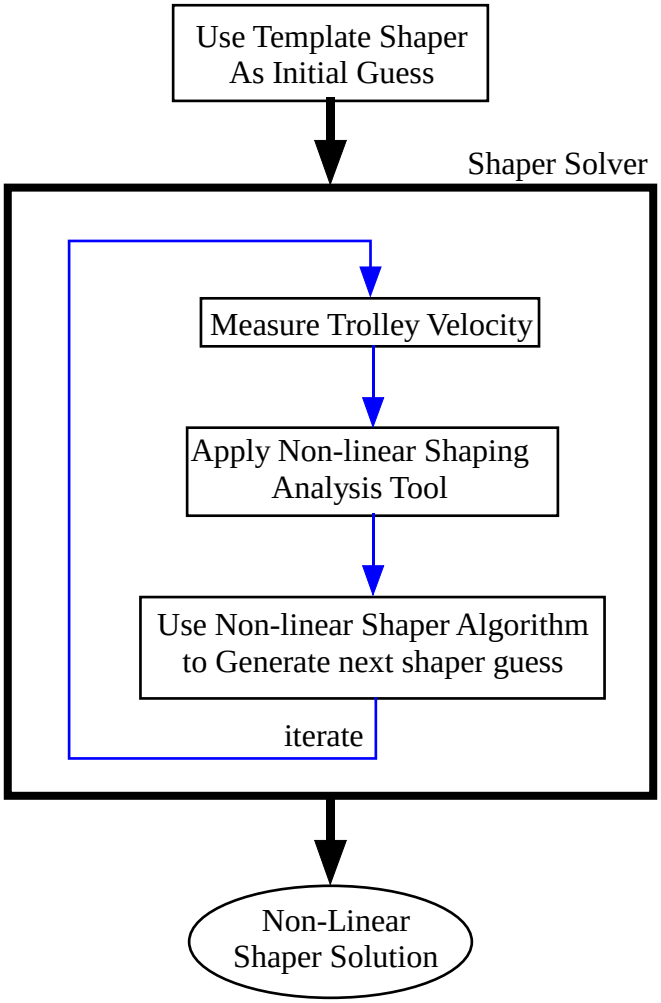
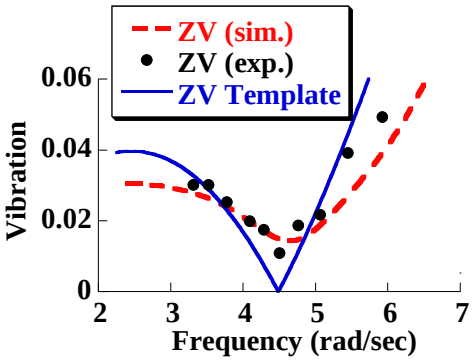


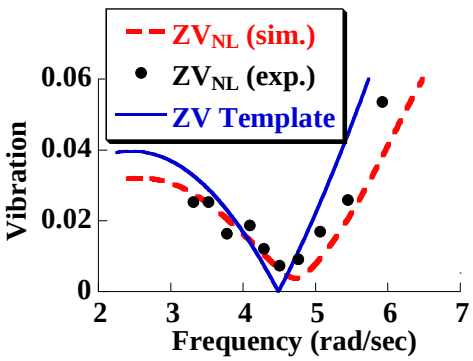
Figure 8.45: Experimental Nonlinear Shaper Solution Process.

The trolley response was measured and the digital analysis tool, presented in Section 8.3.2, was used to derive the amplitude and phase shift of the nonlinear shaper vectors. The nonlinear shaper algorithm was applied to form the next guess for the nonlinear shaper parameters. This process was iterated until the process converged on a solution. In all cases the algorithm converged in 4 iterations or less. The shapers were designed for a suspension length of 0.48 m (19 in) and a maximum trolley velocity of 0.24 m/s.

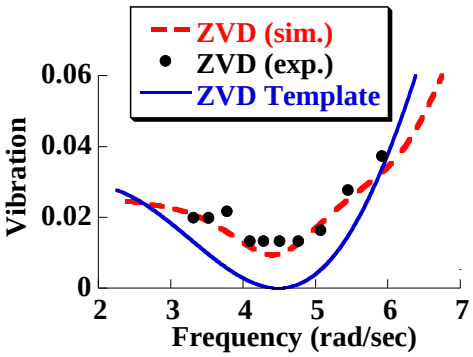
The residual vibration of the shaped commands was measured at different frequencies. This was accomplished by physically changing the payload suspension length and measuring the residual vibration with a digital camera. The theoretical vibration at various frequencies was also determined by simulating the response of a linear, planar crane to the measured



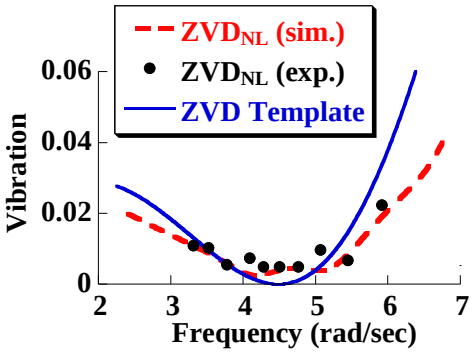
(a) Standard ZV



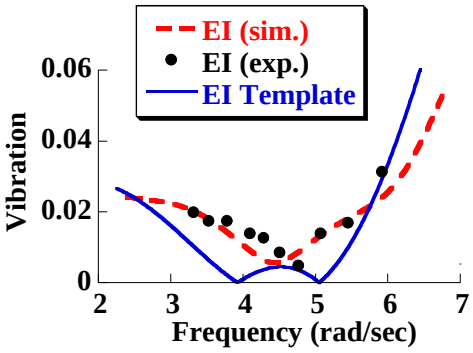
(b) Nonlinear ZV



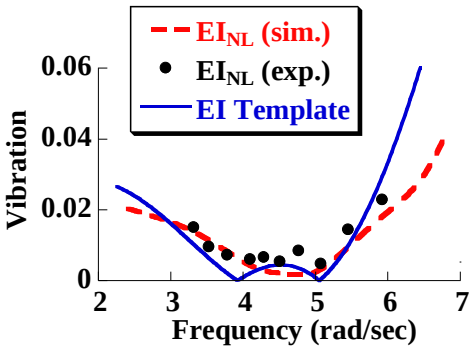
(c) Standard ZVD



(d) Nonlinear ZVD



(e) Standard EI



(f) Nonlinear EI

Figure 8.46: Standard and Nonlinear Shaper Experimental Results.

trolley velocity. These results are shown in Figure 8.46.

Figure 8.46(a) shows the vibration which results when a standard ZV shaper is used on the bridge crane. The solid line shows the template ZV vibration. The dashed line gives the simulated vibration and the black circles show the experimentally measured vibration. The simulation predicts the measured vibration fairly well. Notice the significant discrepancy between the template ZV vibration and the actual crane vibration. This is a direct consequence of the nonlinear dynamics of the drive.

Figure 8.46(b) shows the results for the ZV shaper after the nonlinear shaper algorithm is used to develop a nonlinear ZV shaper for the crane. The new shaper is labeled ZV_{NL}. Again, the dashed-red line shows the simulated vibration and the black circles show the experimentally measured vibration. Comparing Figures 8.46(a) and 8.46(b), the ZV_{NL} shaper is much closer to the ZV template than the standard ZV shaper. In addition, the average vibration amplitude of the standard ZV data is 0.026 radians, whereas the average vibration amplitude of the ZV_{NL} command is 0.021 radians. Using the nonlinear shaper resulted in an 20% reduction in the measured vibration, on average.

Figures 8.46(c) and 8.46(d) show the results for the ZVD shaper, while Figures 8.46(e) and 8.46(f) shows the results for the EI shaper. In both cases the graphs are laid out similar to their ZV counterparts. The improvement from using the nonlinear shaper algorithm is even more pronounced in the ZVD and EI cases than in the aforementioned ZV case. This improvement is seen in both the experimental and simulated data. In both cases the nonlinear shaper matches the template shaper much better than the standard shaper. The average vibration amplitude of the ZVD shaper is 0.02 radians, whereas the average for the ZVD_{NL} is 0.009 radians, a 55% improvement. The average vibration amplitude of the EI shaper is 0.016 radians, whereas the average for the EI_{NL} is 0.01 radians, a 36% improvement.

8.6 Vector Approach: Conclusions

This chapter has presented a series of vector-based, input shaping techniques for nonlinear drive systems. These techniques were developed in both the continuous and digital domains.

The basic idea is to first perform a vector analysis, then use this analysis to design nonlinear compensating input shapers. The vector analysis consists of two basic steps:

1. Divide the drive-system velocity profile into segments.

2. Analyze the amplitude and phase contribution of each segment to the payload response and represent them as vectors.

Once the vector analysis is complete there are two possible strategies for finding useful input shapers. The "Zero-Vibration" strategy is to simply set the resultant from the sum of these vectors equal to zero. This strategy was successfully used to find closed form expressions for UMZV commands for systems with a braking nonlinearity. Simulations and experimental results proved the effectiveness of this approach.

The second shaping strategy is to use a template shaper. The template shaper vectors are related to the nonlinear response vectors through global scaling and rotation factors. This process has the added advantage of imparting some of the frequency response characteristics of the template shaper (eg. frequency robustness) to the resulting nonlinear shaper. This strategy was implemented in two studies. In the first study, the shaper algorithm was applied to a random drive nonlinearity in numerous simulations. In the second study, the shaper algorithm was applied to the experimental bridge crane with low drive gains. In both cases the shaper algorithm not only reduced the residual vibration, but also replicated the frequency characteristics of the template shaper.

Another major contribution of this chapter is a tool for analyzing shaped commands for nonlinear systems. This analysis tool uses the vector analysis ideas and is the nonlinear complement to vector diagrams for linear systems. It visually shows how the nonlinear dynamics degrade the shaper performance in a very specific way. It can be used to "tune" the shaper manually using a trial-and-error process.

The theoretical developments in this chapter were kept as general as possible. Although they were demonstrated with specific cases, the ideas can be easily expanded. For instance, the examples in this chapter used either ZV, ZVD, or EI shapers. But the theory works equally well for multimode shapers, SI shapers, and many more. The theory is also not

limited to crane systems. The plant was assumed to be a planar crane, but any linear system could be used down stream from the nonlinearity. Recall that the basic idea was to separate the nonlinear behavior of the drive, from the linear behavior of the vibratory part of the system. Therefore, this idea can applied to many flexible systems such as robots, spacecraft, and manufacturing equipment.

CHAPTER IX

CONCLUSIONS

Cranes are an integral part of our society. Input shaping has already been shown in improve the efficiency and safety of cranes. The overall goal of this thesis was to make significant strides in the field of input shaping for cranes. This was achieved in five areas:

Experimental Crane Setups Three experimental crane setups were completed: the Hi-Bay crane, the portable bridge crane, and the portable tower crane. These setups were built using state-of-the-art Siemens automation equipment. The setups have been, and continue to be used in several research studies that contribute to this field. In addition, they played an integral role in dynamic systems education at Georgia Tech, Georgia Tech Lorraine, and Tokyo Tech.

Input Shaping Curriculum Development A new approach for teaching input shaping was developed and used in several universities around the world. A complete set of labs was developed that emphasized hands-on learning of input shaping and group learning on an international scale. This laboratory-based curriculum was developed and tested in seven courses, in three different universities, across three different continents. The "large scale" of this effort has led to an extensive set of educational data and a well-tested curriculum. One notable achievement of this process was the student final projects, which are going to be published and presented at three engineering conferences [8, 13, 30].

Input Shaping Operator Studies Input shaping is designed to remove the burden of minimizing payload swing while the operator drives the crane. However, it also changes the crane response to pendant commands. As a result, it will effect how the operator drives the crane. Several studies were conducted to measure how input

shaping effects: path choice, obstacle collisions, learning rate, and remote crane operation. All of these studies showed that crane operators benefitted from using input shaping.

Vector Approach for Designing Shapers for Drive Nonlinearities A new approach for designing input shapers for drive nonlinearities was developed. The effects of the nonlinearity are captured in vectors which can then be manipulated to form shaped commands. This approach was developed in both the continuous and digital domains. In addition, a new tool for analyzing input shaped nonlinear systems was presented. Experimental data and simulations showed the effectiveness of these techniques on cranes. However, most of these developments can be applied to a wide range of nonlinear systems.

Input Shaping for Tower Cranes Several new strategies for designing input shaped commands for tower cranes were developed. First, the related, yet simpler problem of creating shaped commands for a one-mass flyball system was explored. Two strategies emerged from this process that would later be used for tower cranes: linearization and "steady-state relaxation". Then, the tower crane problem was addressed. The equations of motion were derived and further analysis revealed its steady-state condition as well as a 2-mode, linearized model. The linearization and "steady-state relaxation" approaches were then utilized to form new tower crane shapers. The performance of these shapers was compared against linear shapers through simulations and experiments. Both of these techniques can be applied to find shaped commands for other rotating machinery such as boom cranes and robotic arms.

9.1 Future Work

There are several opportunities for future work in the field of input shaping for cranes. Many of the ideas presented in this thesis motivate new studies and research. The work on input shaping for drive nonlinearities has only begun to show the applicability of this new technique. Several more studies can be conducted that target more crane nonlinearities

using the same vector-based approach. Another area of research is refining the digital shaper solver. For example, the solver assumes that both the impulse times and magnitudes of the shaper can be tuned. A new algorithm would be needed for unity magnitude shapers where the impulse magnitudes are fixed. The algorithm also only matches the frequency response of the nonlinear shaper to the template shaper at the model frequency. If more shaper impulses were added the algorithm could be adapted to match the frequency response of the template at multiple points.

The vector-based approach can also be incorporated into a more general shaper design procedure for cranes. The process would adapt the shaper the unique dynamics of each crane, but the overall procedure would stay the same for all cranes. The key steps in this process would be:

1. Gather the desired design specifications (eg. residual vibration constraints, average length, average variation in length, speed requirements).

2. Gather system information about the crane (eg. speed constraints, acceleration constraints, continuous verses on/off speed control).

3. Choose an appropriate linear shaper based on the design specifications and system constraints.

4. Measure the trolley response and use the nonlinear shaper solution to optimize the shaper parameters.

5. If the result does not meet the design constraints, a new linear shaper can be chosen as the initial guess and steps 3-4 are repeated.

If the crane dynamics change over time, the nonlinear shaper solution could be re-applied to "tune-up" the shaper parameters.

The limitations of the vector-based approach can also be examined. This thesis has shown that the technique works in several examples. However, for the technique to be applied to a wider range of systems, including non-crane applications, a more thorough understanding of its limitations is required. For example, the nonlinear braking study

showed that there was no solution for some set of parameters. Perhaps, a more general set of conditions could be derived to determine whether the process will converge on a solution.

More research can be done in the field of input shaping for tower cranes, as well. A more detailed model of the tower crane can be developed to incorporate the dynamics of pulley-type hooks. The model could also be refined to include hoisting. New tower crane shapers could be derived to account for damping or double-pendulum effects using techniques similar to the ones presented in this thesis. The shaping strategies developed for tower cranes can also be extended to several other rotating cranes (eg. boom cranes) and rotating machinery in general (eg. robots, satellites and manufacturing machinery).

The work in education is also an on-going effort. As students and educators use the educational tools developed in this thesis, new areas for improvement will emerge. This might involve revising some of the labs, adding new labs, or modifying the experimental crane setups. The curriculum could also be modified to incorporate more advanced input shaping concepts such as SI shaping, and some of the nonlinear shaping concepts covered in this thesis.

The crane operator studies also suggest many opportunities for further research. All of the studies cited in this thesis could be repeated using more subjects to get more statistically accurate data. Most of the studies focused on comparing input-shaped operation against non-input-shaped operation. However, a whole new set of studies can be performed to compare operator performance for different input shapers. For example, perhaps there are some shapers which are inefficient from a mathematical standpoint, but which are more comfortable for the operator to use and result in more efficient operation. Some of the nonlinear shapers and tower crane shapers developed in this thesis could also be included in these studies.

This thesis has made several contributions to the field of input shaping for cranes. These achievements range from the construction of experimental cranes, to the development of educational tools, to the formulation of new theories on shaper design for tower cranes and nonlinear systems. This work as also paved the way for future studies in these fields. Given the ubiquitous presence cranes in our society, it is the hope of the author that these

developments will contribute not only to the academic community but society at large.

APPENDIX A

JAPAN/US FINAL PROJECTS

The final project description for the 2005 Japan/US input shaping class appears below:

ME6404 Final Project
Assigned: Friday, 10/28/2005

Your team will formulate and complete a control system implementation project in one of the three areas described below. The project should have some component that is completed on a crane; however, a significant component of the project can be performed in simulation. The project can be based on any control method, but the techniques should be somewhat related to the material in this course. The project should have the following components:

1) Clear statement of system and objectives
2) Development of a system model
3) Design of a control system
4) Robustness analysis of the control system
5) Hardware verification

You will choose your own teammates for this project. The teams will be composed of three students. Those not forming their own team will be randomly assigned to a team.

Due: Wednesday, 2 November at the beginning of lecture:
A short proposal describing your project. You will present this to the class and the class will have a short brainstorming session on your proposal to suggest problems and possible solutions. The proposal should be at most 1/2 page of text and 1 or 2 figures. **Put the proposal on 1 overhead transparency and send a pdf version to Dr. Singhose.**

Due: Monday, Wednesday 30 November in lecture:
1) Short presentation of your project (5 minutes) in Powerpoint or on overhead slides. You will give this presentation in the lecture room and you will be evaluated by the other students in the class.

2) Written report on your project. Be concise and only include figures that demonstrate an important result. Insert the figures into the text near where they are first cited. **The report is limited to 10 pages, including figures.** Use 1" margins on all sides and use a 12pt. font size. Place computer code in an Appendix. These pages do not count against your 10 pages. However, do not attempt to place important information in an Appendix to circumvent the 10 page limit.

Possible Projects Areas: There are three teams, so please choose one of the following three areas for your project. If more than one team wants to work on one of the suggested areas, then a Brian Fatkin induced coin toss will decide which team gets their first choice.

1) Remote Operator Study: Remote operation of cranes is a necessity in hazardous environments. However, it is often more difficult for crane operators to maneuver the payload when it is remote. Possible causes for this include a poor visual interface and communication delays. This project will examine this phenomenon and explore what factors effect operator performance. Possible project components include:
   - Create two identical obstacle courses for the bridge crane in Atlanta and the tower crane in Japan. Run studies to compare operator performance on the remote verses non-remote courses.
   - Create different obstacle courses and try to determine what types of obstacle arrangements are particularly challenging.
   - Effect of motion parameters. Adjust the speed and/or acceleration and explore how operator performance is affected (eg. Operation may be easier with slower acceleration), but the task may take longer.
   - Effect of input shaper. Try to develop an input shaper specifically tailored to remote operation.
   - Effect of a secondary oscillation due to payload dynamics. Add a payload that causes a secondary oscillation. How does this effect manipulation? How can this be counteracted?

2) Dynamic Characterization of Shaped and Unshaped Crane Responses: A crane is typically modeled as a linear system. However, this approximation only holds for small deflection angles, certain types of motion, and certain types of cranes. For example, consider the bridge crane shown in Figure 1.

The equations of motion are given by:

$$L\ddot{\phi} + L\dot{\theta}^2\cos(\phi)\sin(\phi) + g\sin(\phi)\cos(\theta) = \ddot{x}\cos(\phi) + \ddot{y}\sin(\phi)\sin(\theta)$$
$$L\ddot{\theta}\cos(\phi) - 2L\dot{\phi}\dot{\theta}\sin(\phi) + g\sin(\theta) = -\ddot{y}\cos(\theta)$$

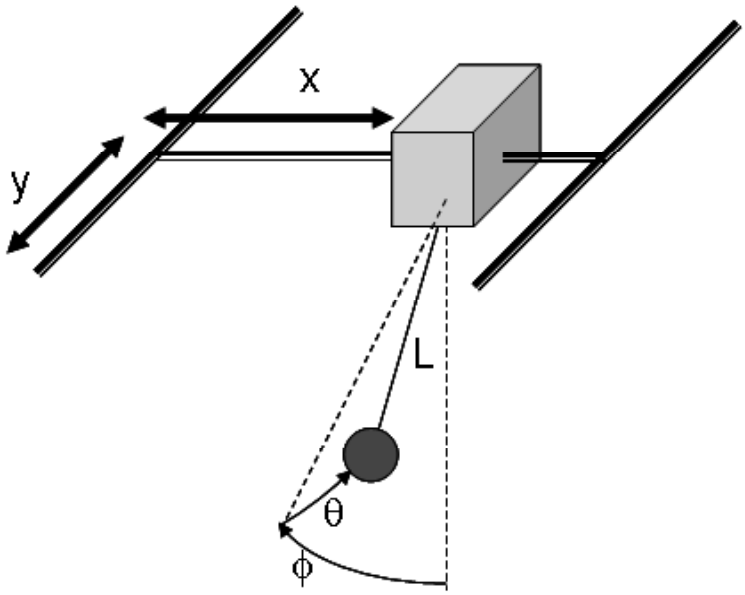


Figure 1: Bridge Crane.

To transform the payload location to fixed coordinates use the transformation:

$$x_{payload} = x - \mathrm{L}\sin(\phi)$$
$$y_{payload} = y + \mathrm{L}\cos(\phi)\sin(\theta)$$

As a second example, consider the tower crane shown in Figure 2. Its equations of motion are given by:

$$\begin{aligned}\mathrm{L}\ddot{\phi} + \mathrm{L}\dot{\theta}^2\cos(\phi)\sin(\phi) + g\sin(\phi)\cos(\theta) = {} & -\ddot{\mathrm{R}}\cos(\phi) + \mathrm{R}\dot{s}^2\cos(\phi) - \mathrm{R}\ddot{s}\sin(\phi)\sin(\theta) - 2\dot{\mathrm{R}}\dot{s}\sin(\phi)\sin(\theta) \\ & - 2\mathrm{L}\dot{s}\dot{\theta}\cos^2(\phi)\cos(\theta) - \mathrm{L}\ddot{s}\sin(\theta) + \mathrm{L}\dot{s}^2\sin(\phi)\cos^2(\theta)\cos(\phi) \\ \mathrm{L}\ddot{\theta}\cos(\phi) - 2\mathrm{L}\dot{\phi}\dot{\theta}\sin(\phi) + g\sin(\theta) = {} & \mathrm{R}\ddot{s}\cos(\theta) + 2\dot{\mathrm{R}}\dot{s}\cos(\theta) + 2\mathrm{L}\dot{s}\dot{\phi}\cos(\phi)\cos(\theta) + \mathrm{L}\ddot{s}\sin(\phi)\cos(\theta) \\ & + \mathrm{L}\ddot{s}\sin(\phi)\cos(\theta) + \mathrm{L}\dot{s}^2\sin(\theta)\cos(\phi)\cos(\theta)\end{aligned}$$

To transform the payload location to fixed coordinates use the transformation:

$$x_{payload} = \cos(s)(\mathrm{R} + \mathrm{L}\sin(\phi)) + \mathrm{L}\sin(s)\sin(\theta)\cos(\phi)$$
$$y_{payload} = \sin(\mathrm{s})(R + \mathrm{L}\sin(\phi)) - \mathrm{L}\cos(\mathrm{s})\sin(\theta)\cos(\phi)$$

In both cases the equations of motion are clearly nonlinear. Recall that input shapers are designed for linear systems. Therefore, input shaping will become less effective for cases where bridge and tower cranes perform in a highly non-linear manner.

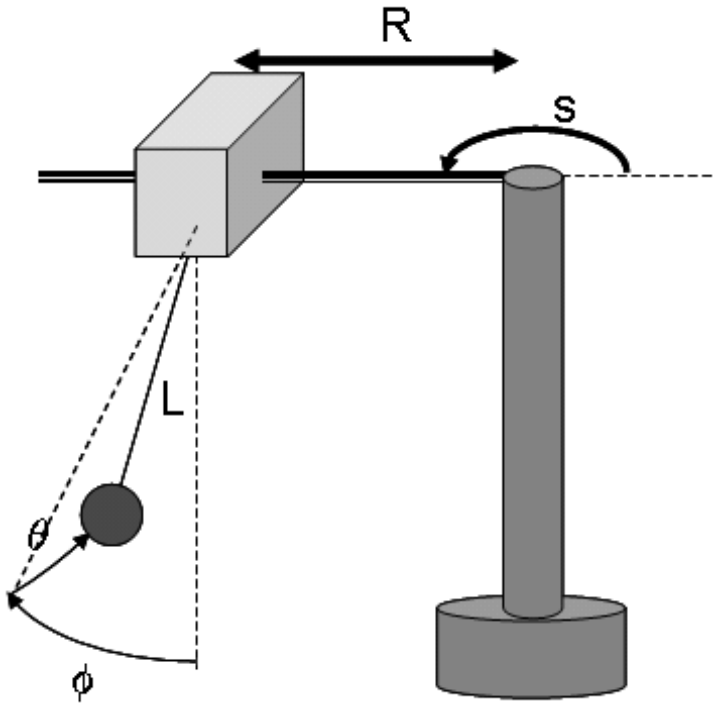


Figure 2: Tower Crane.

Possible project components include:

- Show that the given equations of motion are valid for both the bridge crane in Atlanta and the tower crane in Tokyo.
- Determine when the system transitions from behaving linearly to non-linearly.
- Examine how the transition to nonlinear behavior affects the performance of a simple input shaper.
- Design advanced input shapers to deal with nonlinearities and experimentally verify your predictions.

3) Drive System Nonlinearity: Most real systems, while modeled as linear, contain non-linear elements. For example, consider the velocity response of a crane trolley to a pulse command shown in Figure 3. Notice that when the system acclerates and decelerates, it approaches the final setpoint using an exponential (first-order) profile. However, the relationship between the desired and smoothed command is non-linear because the acceleration and deceleration time constants are different.

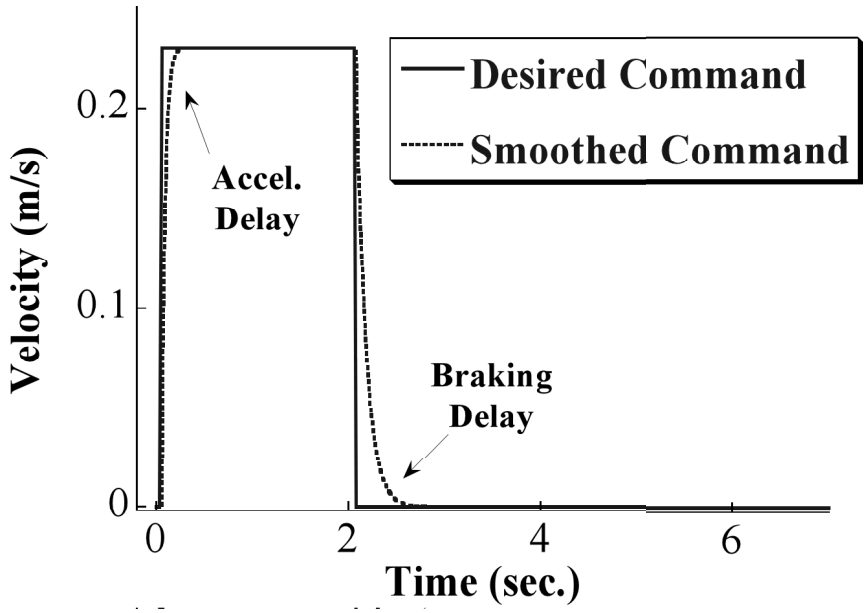


Figure 3: Drive System Response.

The above nonlinearity has detrimental effects on input shapers. Some shapers are not effected very much, while others are more susceptible to this effect. For example, consider applying a UMZV shaper to the above desired command. The results are shown in Figure 4.

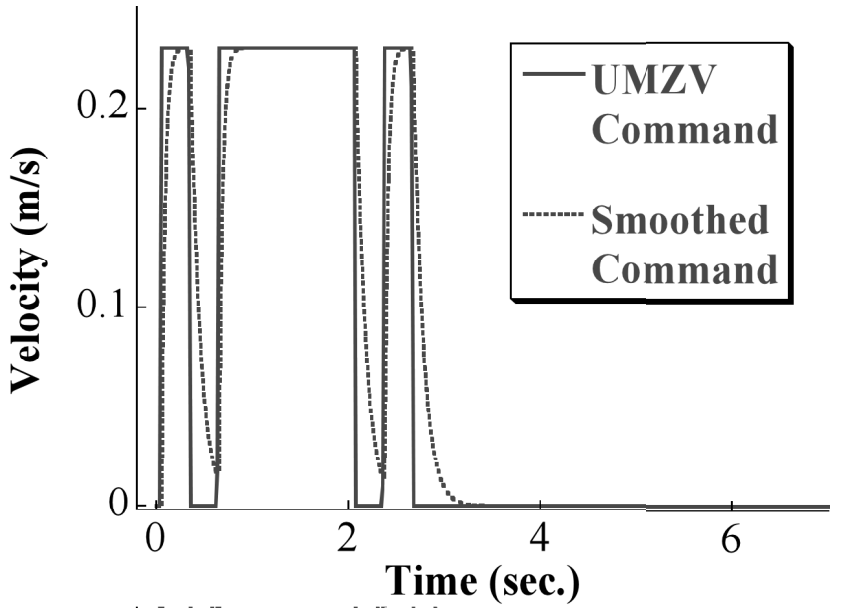


Figure 4: Drive System Response to UMZV Shaper.

If this command were applied to a real crane, the payload would not have zero residual vibration.

A new UMZV shaper has been developed to compensate for this nonlinear effect. The impulse times and amplitudes of this new shaper are given by:

$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 \\ 0 & t_2 & t_3 \end{bmatrix}$$

where,

$$t_2 = \frac{1}{\omega_n} * \left(\cos^{-1}\left(\frac{\tau_b \sqrt{\omega_n^2 + \tau_a^2}}{2\tau_a \sqrt{\omega_n^2 + \tau_b^2}} \right) - \tan^{-1}\left(\frac{\omega_n}{\tau_b} \right) + \tan^{-1}\left(\frac{\omega_n}{\tau_a} \right) \right)$$

$$t_3 = \frac{2}{\omega_n} * \cos^{-1}\left(\frac{\tau_b \sqrt{\omega_n^2 + \tau_a^2}}{2\tau_a \sqrt{\omega_n^2 + \tau_b^2}} \right)$$

The derivation of this new shaper can be supplied if necessary.
Possible project components include:

- Predict and verify the effect of this type of drive system nonlinearity on standard positive shapers.
- Predict and verify the effect of this type of nonlinearity on negative input shapers.
- Explore the effectiveness of the new shaper for various acceleration and deceleration time constants.
- Explore the effectiveness of this shaper for multimode systems.
- Examining the sensitivity of this shaper to system variations.

APPENDIX B

FINAL LAB MODULES

The 5 finalized lab modules appear below:

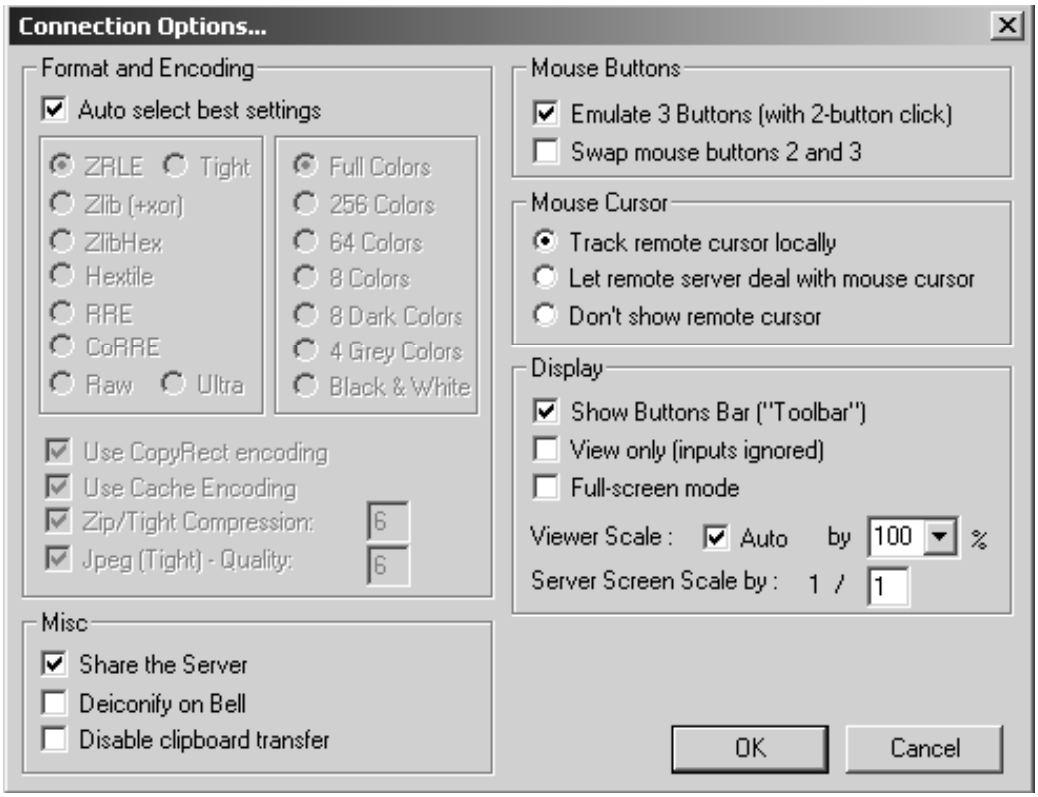
Module 0:
Logging on and Basic Crane Operation

Install UltraVNC:

1. Goto: http://ultravnc.sourceforge.net/
2. Click on “Ultr@VNC 1.0.1 Standalone Viewer” on the right of the screen
3. Select a site and download the file
4. Unzip the file
5. Copy vncviewer.exe to the desktop

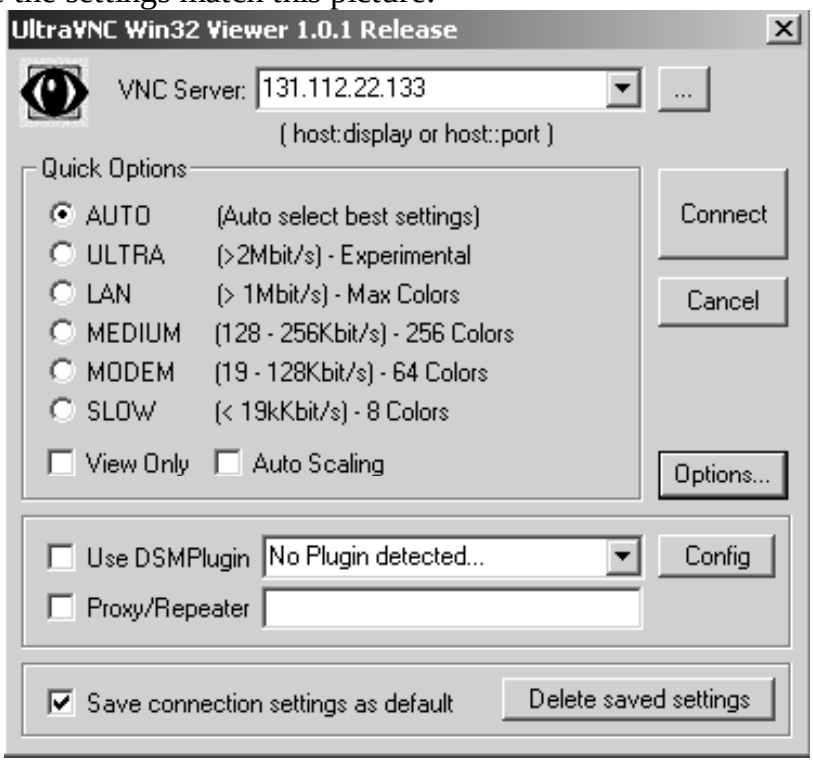
Running UltraVNC:

1. Double-click on vncviewer.exe
2. Click the Options Button. Make sure the settings match the picture below:



3. Click “OK”

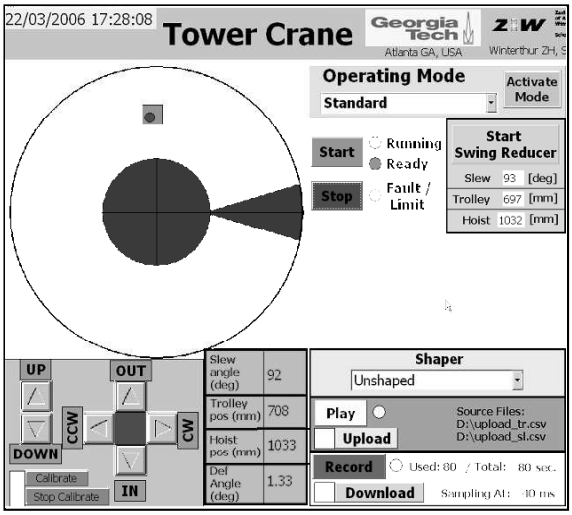
4. Make sure the settings match this picture:



5. Click Connect
6. When prompted for the password type: todai06
7. You now have control of the computer the runs the crane

Starting the Programs:

1. Double click the icon labeled “Web Cam”. This shows real-time video of the tower crane.
2. Go back to the desktop and double click the icon labeled “Tower Crane”. This will bring up the Tower Crane controls:



3. Reduce the window size so that the webcam screen is visible. However if the connection is too slow close the webcam window.

Basic Crane Operation:

1. The "Operating Mode" box, in the top right corner of the screen, is used to change the operating mode for each lab module. For now select **Standard** and press the "Activate Mode" button.
2. Beneath the "Operating Mode" box are the "Start" and "Stop" buttons. **Push the "Start" button** now to turn on the power. Notice that the indicator to the right of these buttons has moved from "Ready" to "Running". If you ever need to execute an emergency stop, push the "Stop" button.
3. Beneath the "Stop" button is the "Shaper" selection box. **Select Unshaped.** In future labs you will choose various Input Shapers (eg. ZV, ZVD, etc…) using this menu.
4. The crane is controlled using the directional buttons on the lower left of the screen. "UP" and "Down" will raise and lower the payload. "CW" and "CCW" will rotate (slew) the crane clockwise and counter-clockwise respectively. "In" and "Out" will move the trolley inward and outward in a radial motion. **Experiment with moving the crane now**.
5. As you move the crane notice that the upper left corner shows a real-time animation of the crane's configuration. The green-box represents the trolley position and the red-circle is the payload position.
6. As you move the crane, also notice the position information indicated in the blue-box in the bottom-middle of the screen. This shows the trolley position in polar coordinates, as well as the payload height and deflection.
7. The black areas of the animated sketch indicate the limits of the crane's motion. If you ever run into these limits the crane will automatically stop. You will then only be allowed to move the crane away from that limit.

Using the Swing Reducer:

1. In many of the labs it will be important to zero-out the payload swing before running trials. This is done with the "Swing Reducer" button in the top left. **Push the "Start Swing Reducer" button now.** The begin moving on its own to zero-out the swing. You cannot move the crane manually while the swing reducer is engaged.
2. Underneath the "Swing Reducer" button are 3 entry boxes for the slew position, trolley position, and hoisted height. These numbers are the desired, steady-state position of the crane. The swing reducer will eliminate the swing and return the crane back to this position. Notice that when you initially pushed the "Start Swing Reducer" button the current position of the crane was instantly copied into these boxes.
3. If you want to move the crane automatically to a new position, with no swing, enter the corresponding coordinates in the swing reducer boxes. For example, **enter a new slewing position (in degrees) in the slew box.** As soon as you press enter the crane will move to the new desired position. This only works while the "swing reducer" button is depressed.

Record and Playback Buttons:

1. In the bottom right of the screen are the “record” and “play” buttons.
2. The “play” button is used to automate the crane. A series of velocity setpoints can be created in Xcell and stored in a *.csv file using the filenames shown on the screen. This file can be loaded into the system by pressing the “upload” button. Then the setpoints can be played back automatically using the “play” button.
3. The “record” button will be used in every lab. It is used to record the position data of the crane and payload.
4. **Push the “record” button now.** You will notice that the recording light turns on and the recording timer starts counting. When the counter reaches the limit shown on the screen it will automatically stop. The sampling rate is also shown. You can manually stop the recording before it reaches the limit by pushing the recording button again. **Push the record button now to stop recording.**
5. After you finish recording the data must be downloaded to the computer. This is accomplished by pushing the “download” button.

Ending the session:

1. When you finish driving the crane **press the “Stop” button**. It is important that you turn off the crane after every lab session so it does not stay on for long periods.
2. Then close the UltraVNC window on your screen. **Do this now.**

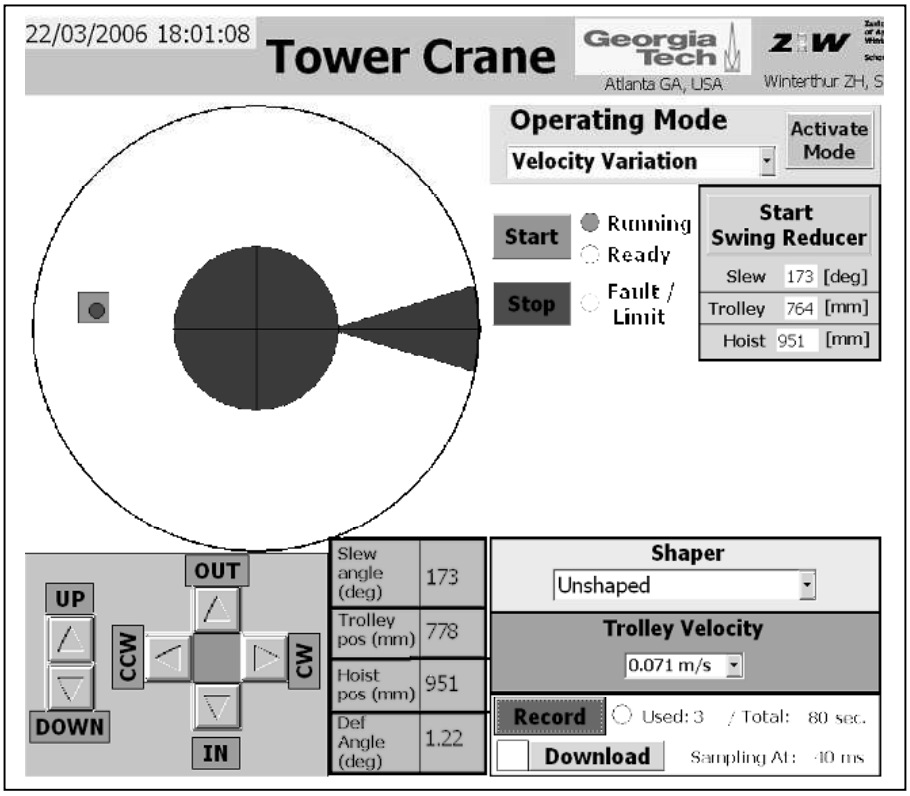
Module 1:
Velocity Variation

Objective: Move the trolley a set distance while varying the velocity. Record the effect on the payload oscillation.

Procedure:

Initialize the Velocity Variation Mode

1. Select "Velocity Variation" from the Operating Mode box at the top. Then push "Activate Mode".
2. The control screen should look like this:



Perform the Experiment:

1. Move the crane with zero swing to an initial position of: trolley=550 mm (this can be off by ±10 mm), hoist=1200 mm. The slewing position does not matter.
2. Stop the swing reducer and select "Velocity Variation" from the "Shaper" drop-down menu.
3. Select the lowest velocity (0.071) in the "Trolley Velocity" drop down menu.
4. Push the "Record" button to begin recording data.
5. Hold down the "Out" button to make the trolley move radially outward. The trolley will automatically stop after it moves 300 mm. Keep the button held down until the payload has swung back-and-forth 3 times.

6. Release the "Out" button. This will cause the trolley to return back to its starting location.
7. Push the "Record" button to stop recording data.

Collecting the data

1. Push the "Download" button.
2. Launch MATLAB from the desktop
3. Type "Velocity_Variation" and press enter.
4. It will ask for a name. Enter a file name where you want to store the data. The data is stored in "d:student_data/Velocity_Variation" as a text file. The data is stored as:

time (s)	Trolley Radial Position (m)	Trolley Radial Velocity (m/s)	Payload Radial Position (m)

5. The program also plots your results

Finish Collecting Data:

- Rerun the test using the other velocity values in the "Trolley Velocity" drop down menu.

Post-Lab Analysis:

- Plot the residual vibration amplitude vs. the trolley velocity. The vibration should be measured after the trolley has stopped moving.

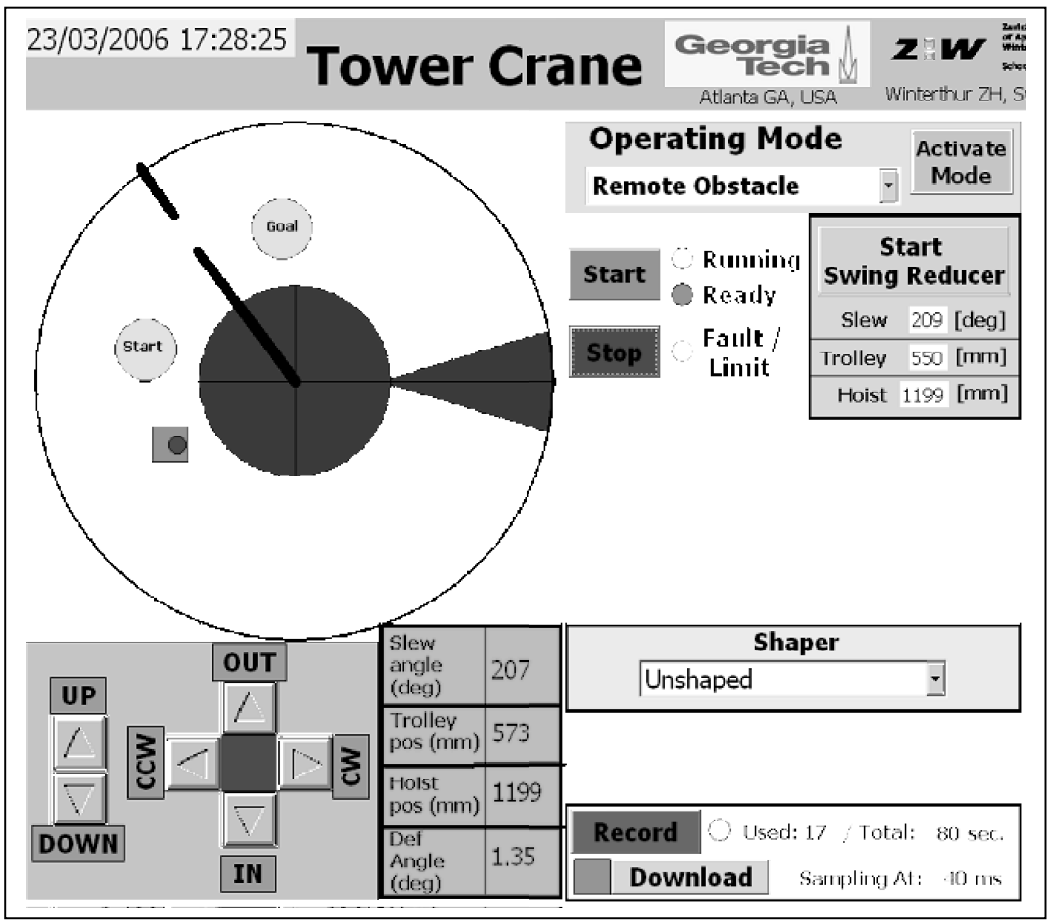
- Explain the shape of your curve.

Module 2:
Remote Obstacle Course Driving

Objective: Drive the tower crane through the obstacle course remotely from your computer. Drive with and without Input Shaping and compare the results.

Procedure:

- Select "Remote Obstacle" from the Operating mode menu at the top right of the screen. Press "Activate Mode". The screen will look like this:



Drive the Crane through the Obstacle Course without Shaping:

1. Select "Unshaped" under the "Shaper" Menu
2. Move the crane using the swing reducer to the start position: slew=170^0, trolley=650 mm, Hoist=1300mm
3. Turn off the swing reducer and push "Record".
4. Drive the crane from the "Start" to the "Goal". Try to avoid the obstacles. Wait until the payload (red circle) is entirely contained within the "Goal" circle. Then press the "Record" button to stop recording data. However, if 80 seconds has passed the system will stop recording automatically and you must re-do the trial.

Collecting the data:

1. Push “Download”
2. Go to the desktop and start “MATLAB 5.3”
3. Type: “Remote_Obstacle”
4. It will ask for a name. Enter a file name where you want to store your data. The data is stored in “d:student_data/Remote_Obstacle/” as a text file. The data is stored as:

time (s)	Trolley Rotational Position (Deg)	Trolley Radial Position (m)	Payload Rotational Position (Deg)	Payload Radial Position (m)

5. The program also plots your results
6. Save the MATLAB figure file

Repeat with Input Shapin:g

- Repeat the procedure with ZV shaping. Select “ZV1300” from the “Shaper” drop down menu. Then repeat the above procedure.

Post-Lab Analysis:

- Compare the results from your Shaped and Unshaped runs.

Email your two data files to: **jason.w.lawrence@gmail.com**

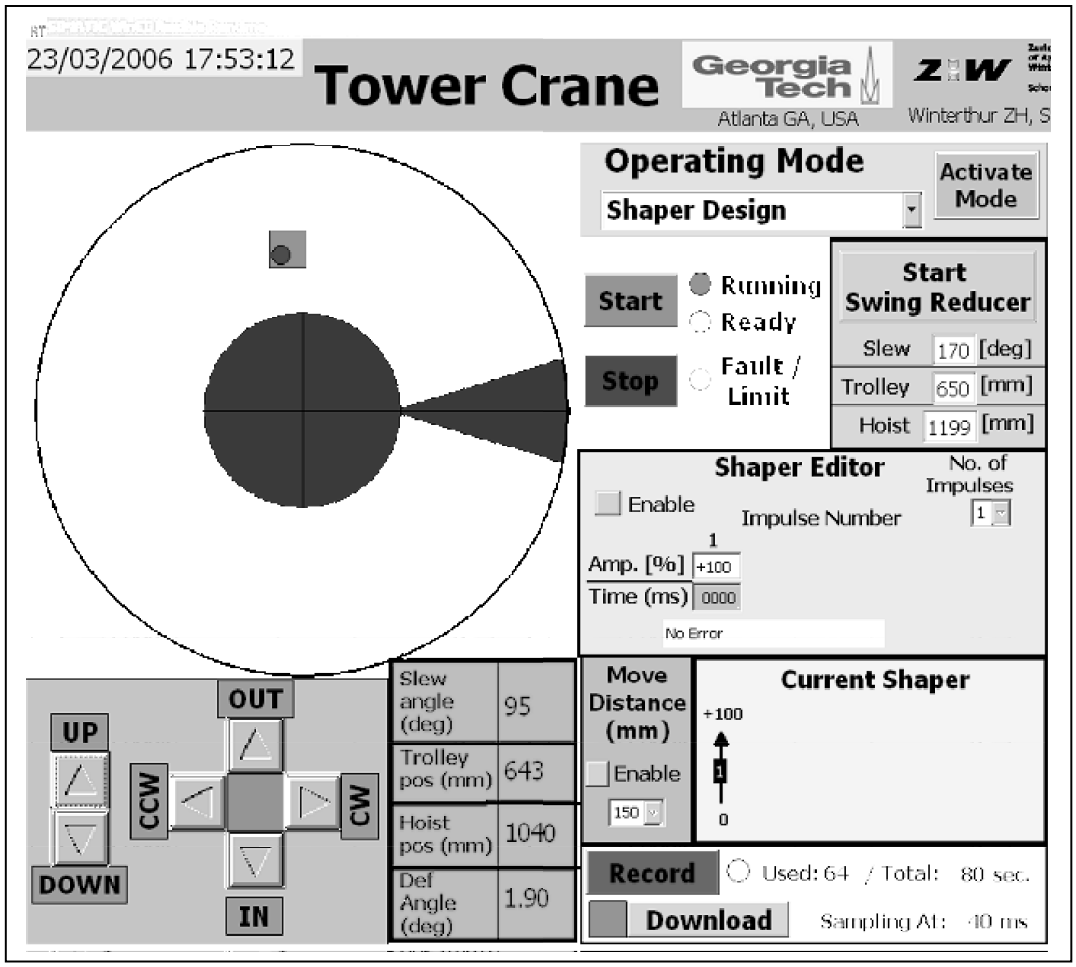
Module 3:
Input Shaper Design and Robustness

Objective: The overall objective is to design input shapers and measure their performance. There are three subgoals:

1. Design ZV and ZVD shapers from frequency and damping ratio data.
2. Compare how move distance effects residual vibration for ZV shaped and unshaped motions.
3. Compare the robustness of ZVD and ZV shapers to variations in frequency

Procedure:

1. Select "Shaper Design" from the Operating mode menu at the top right of the screen. Press "Activate Mode". The screen should look like this:



2. Make sure the "enabled" box is unchecked in the "Shaper Editor" and "Move Distance" areas. This will allow you to drive the crane normally.

Part 1 (Move distance Tests):

3. Using the swing reducer, move the crane without swing to the starting position: trolley=530 mm (this position can be off by ± 10 mm), the hoisted position can be anywhere between 700 and 900, the slewing position does not matter.
4. In the "Move Distance" area click the "enable" box and select 100 mm from the drop down menu
5. Click the "Record" button to being recording data
6. Press and hold the "Out" button. The trolley will move 100 mm and automatically stop while you hold the button. When you release the button the trolley will return to its starting point.
7. Press the "Record" button again to stop recording

Collecting the data

1. Push the "Download" button.
2. Launch MATLAB from the desktop
3. Type "Shaper_Design" and press enter.
4. It will ask for a name. Enter a file name where you want to store the data. The data is stored in "d:student_data/Shaper_Design/" as a text file. The data is stored as:
5.

Time (sec)	Trolley Commanded Velocity (m/s)	Trolley Measured Velocity (m/s)	Payload Radial Swing (deg)

6. The program also plots your results. You can measure the payload swing amplitude directly from these plots if you wish.

More Tests

Repeat Part 1 using all of the distances in the "Move Distance" drop down menu.

ZV Input Shaping

1. Design a ZV input shaper for the hoisted height of the crane. Assume that the damping ratio of the system is zero.
2. In the "Shaper Editor" box select "enable". Select "No. of impulses" to be 2. Then, type in the amplitude and times of your shaper impulses. Note that the impulses values are entered as a percent (between –100%…100%) and the impulse times are entered in milliseconds.
3. The "Current Shaper" box displays the input shaper currently being used by the system. If your shaper values are possible they will be automatically entered into the system and will appear in the "Current Shaper" box. If they are not possible (eg. Impulses don't add to 1, impulse sum exceeds 100% at any point, or the impulse times are not ordered) then an error will be displayed.
4. Repeat the move distance tests from the previous section.

Analysis

- For each trial measure the peak-to-peak swing amplitude at the completion of the motion.
- Plot the swing amplitude vs. move distance for the unshaped case
- Plot the swing amplitude vs. move distance for the ZV shaped case
- Compare the two plots

Part 2 (Shaper Robustness):

1. Using the "Swing Reducer" move the crane without swing to the starting position: trolley pos=530 mm (this position can be off by ± 10 mm), hoisted pos=1600 mm, the slewing position does not matter.
2. In the "Move Distance" area make sure the "enable" box is selected. Choose 150 mm from the drop down menu.

ZV Shaped Tests

1. Derive a ZV shaper for a hoisted length of 1000 mm. This shaper will be used for all of the ZV tests.
2. Enter this ZV shaper in the "Shaper Editor" using the procedure from Part 1.
3. Use the same procedure from Part 1 to record the swing for all the available move distances (150mm..350mm).
4. Repeat these tests for the following hoisted lengths: 1000 mm, 600 mm

ZVD Shaped Tests

5. Derive a ZVD shaper for a hoisted length of 1000 mm. This shaper will be used for all of the ZVD tests.
6. Repeat the procedure from the ZV Shaped tests.

Analysis

- For each trial measure the peak-to-peak swing amplitude at the completion of the motion.
- Plot the swing amplitude vs. hoisted length for the ZV cases
- Plot the swing amplitude vs. hoisted length for the ZVD cases
- Compare the two plots

Module 4:
Automated Obstacle Course Navigation

Objective: The objective is to use input shaping to create an trajectory that moves the crane through an obstacle course quickly and without colliding into obstacles.

Lab Procedure

Enter the Input Shaper

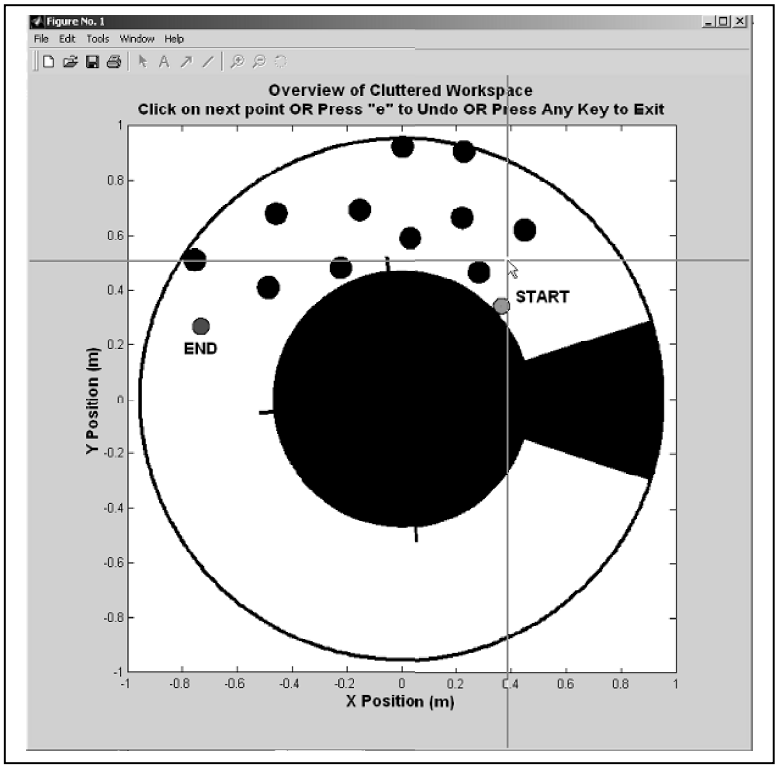
1. Choose a payload height between 1400mm and 1000 mm. Write down the number you have selected
2. Design a input shaper for the selected payload height.
3. Create a text file in "d:student_data\auto_obstacle\" of the tower crane laptop. In this text file, enter the impulse times and amplitudes of the Input Shaper you designed. The format is:

   [time (s) | Amplitude (value)]

   Use the file "input_shaper_template.txt" as a temple for this file.

Generate the Trajectory:

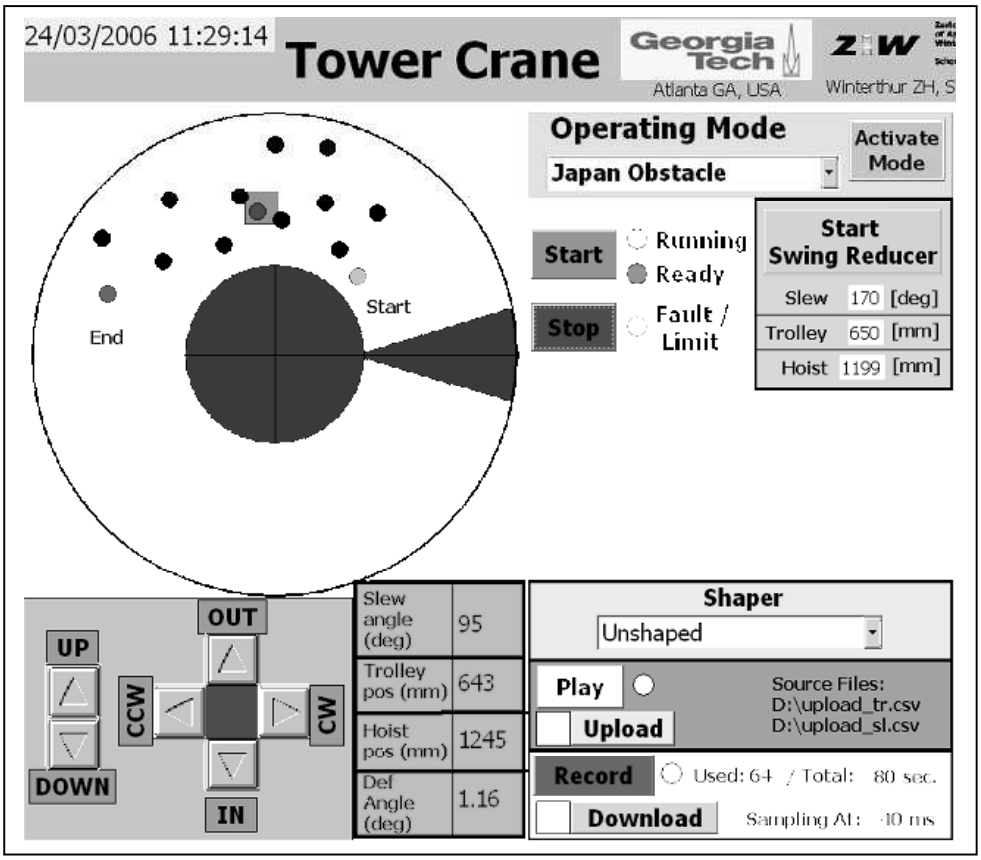
1. Start matlab and maximize the window for better viewing.
2. Type "auto_obstacle" and press enter.
3. Type "1" and press enter to generate a new trajectory.
4. A figure will appear that shows the obstacle course:



5. To create a path for the trolley, click on points in the figure. After each point, enter a time period that you want to wait at that location. **Hint: Adding a wait time may reduce swing and warping of the shaped trajectory, but it slows down the system.** Press enter and it will ask for a new trajectory point.
6. To erase or Undo a point press "e" while the figure is visible.
7. When you are done making the path, press any key to exit while the figure is visible.
8. The computer will then ask you to enter the filename of the input shaper you saved in "d:student_data\auto_obstacle\". Enter the full name and press enter.
9. Matlab will then plot two figures. The front-most figure (figure 2) shows the original trajectory you entered with the mouse and the input-shaped version of the trajectory. Minimize this figure. Behind it you will see figure 1. The plots the original and shaped position and velocity in both Cartesian and polar coordinates.

Run the Trajectory

1. Goto the desktop and launch the "Tower Crane" interface.
2. In the upper left corner use the drop down menu. Select "Japan Obstacle" and then push the "Activate Mode" button. The screen will look like this:



3. Use the "swing reducer" to move the payload without swing to the start position: slew=42, trolley=519. In the "Hoist" field enter the height you chose in the beginning of the lab.

4. Push the UPLOAD button and wait till the Upload light stops flashing
5. Push RECORD then push PLAY. The crane will play your trajectory.
6. When it is finished push "STOP RECORD" and "STOP PLAYBACK".
7. Push the DOWNLOAD button.

Retrieving the Data:

1. Go back to Matlab and type: "auto_obstacle" and press enter.
2. Type "2" and press enter to load and view the data you recorded.
3. Enter a filename to save you data. The data stored is in the format:

time (s)	Trolley Rotational Position (Deg)	Trolley Radial Position (m)	Payload Rotational Position (Deg)	Payload Radial Position (m)

4. Matlab will also plot the Trolley and Payload positions.

Post-Lab Questions:

1. What type of input shaper did you use? Why did you choose this shaper? How did you choose the trajectory?

APPENDIX C

ZV2$_{\text{LIN}}$ POLYNOMIAL COEFFICIENTS

C.1 Rising Command

The rising portion of the command is described by the values t_2, t_3, A_1, A_2, A_3 as shown in Figure 7.18. Table C.1 below gives a coefficients to a polynomial curve fit for each of the values. The coefficients are used as follows:

$$f(L, \dot{s}_f) = c_1 + c_2 L + c_3 \dot{s}_f + c_4 L \dot{s}_f + c_5 L^2 + c_6 \dot{s}_f^2 + c_7 L^3 + c_8 L^2 \dot{s}_f + c_9 L \dot{s}_f^2 + c_{10} \dot{s}_f^3$$

This polynomial fit is accurate for suspension lengths $L = [10 \ldots 30]$ m and slew velocities $\dot{s} = [0.05 \ldots 0.1]$ rad/s. Note that A_3 is not in the chart because it can be calculated from: $A_3 = 1 - (A_1 + A_2)$.

Table C.1: Coefficients for Rising Portion of the ZV2$_{\text{Lin}}$ Command for $L = [10 \ldots 30]$ m and $\dot{s} = [0.05 \ldots 0.1]$ rad/s.

	A_1	A_2	t_2	t_3
c_1	0.3407	0.4924	1.3686	2.7951
c_2	0.0001	0.0002	0.2303	0.4727
c_3	0.0346	0.0655	0.0392	0.2512
c_4	-0.0018	-0.0041	0.0182	-0.0191
c_5	0.0000	0.0000	-0.0041	-0.0084
c_6	-0.1857	-0.3088	0.0817	-1.3122
c_7	0.0000	0.0000	0.0000	0.0001
c_8	0.0000	0.0000	-0.0007	0.0001
c_9	0.0276	-0.0178	-0.2752	0.1064
c_10	-0.3250	-0.6220	1.1985	-1.9729

C.2 Falling Command

The falling portion of the command is described by the values t_4, t_5, A_4, A_5, A_6 as shown in Figure 7.18. Table C.2 below gives a coefficients to a polynomial curve fit for each of the values. This polynomial fit is accurate for suspension lengths $L = [10 \ldots 30]$ m and slew velocities $\dot{s} = [0.05 \ldots 0.1]$ rad/s. Note that A_6 is not in the chart because it can be calculated from: $A_6 = 1 - (A_4 + A_5)$.

Table C.2: Coefficients for Falling Portion of the ZV2_{Lin} Command for $L = [10 \ldots 30]$ m and $\dot{s} = [0.05 \ldots 0.1]$ rad/s.

	A_4	A_5	t_3	t_4
c_1	0.1617	0.4924	1.4279	2.7977
c_2	0.0001	0.0002	0.2425	0.4727
c_3	0.0187	0.0649	0.1900	0.2046
c_4	-0.0015	-0.0042	-0.0406	-0.0201
c_5	0.0000	0.0000	-0.0043	-0.0084
c_6	-0.0765	-0.3033	-1.4002	-1.2588
c_7	0.0000	0.0000	0.0000	0.0001
c_8	0.0000	0.0000	0.0010	0.0003
c_9	0.0289	-0.0086	0.4425	0.1746
c_10	-0.1272	-0.6048	-3.0539	-1.6616

APPENDIX D

UMZV2$_{\text{LIN}}$ POLYNOMIAL COEFFICIENTS

D.1 Rising Command

The rising portion of the command is described by the values $t_2, \ldots, t_5$ as shown in Figure 7.22. Table D.3 below gives a coefficients to a polynomial curve fit for each of the values. The coefficients are used as follows:

$$f(L, \dot{s}_f) = c_1 + c_2 L + c_3 \dot{s}_f + c_4 L \dot{s}_f + c_5 L^2 + c_6 \dot{s}_f^2$$

This polynomial fit is accurate for suspension lengths $L = [10 \ldots 30]$ m and slew velocities $\dot{s} = [0.05 \ldots 0.1]$ rad/s. Note that t_1 is assumed to be zero.

Table D.3: Coefficients for Rising Portion of the UMZV2$_{\text{Lin}}$ Command for $L = [10 \ldots 30]$ m and $\dot{s} = [0.05 \ldots 0.1]$ rad/s.

	t_2	t_3	t_4	t_5
c_1	0.4231	1.1785	2.3202	2.5132
c_2	0.0438	0.1288	0.2470	0.2670
c_3	-0.1570	-0.0267	-0.4809	-0.5481
c_4	0.0293	0.0193	0.1010	0.1171
c_5	-0.0004	-0.0011	-0.0021	-0.0023
c_6	-0.1727	-0.6116	-0.8791	-1.0228

D.2 Falling Command

The falling portion of the command is described by the values $t_6, \ldots, t_9$ as shown in Figure 7.22. This polynomial fit is accurate for suspension lengths $L = [10 \ldots 30]$ m and slew velocities $\dot{s} = [0.05 \ldots 0.1]$ rad/s. Table D.4 below gives a coefficients to a polynomial curve fit for each of the values.

Table D.4: Coefficients for Falling Portion of the UMZV2$_{\text{Lin}}$ Command for $L = [10 \ldots 30]$ m and $\dot{s} = [0.05 \ldots 0.1]$ rad/s.

	t_6	t_7	t_8	t_9
c_1	0.1831	1.5618	2.2468	2.7761
c_2	0.0204	0.1406	0.2293	0.2722
c_3	0.0684	-3.8568	-2.5976	-4.3194
c_4	0.0080	0.0472	-0.0299	0.0156
c_5	-0.0002	-0.0013	-0.0021	-0.0025
c_6	-0.2345	7.6500	4.9419	7.9345

APPENDIX E

EDUCATION FOLLOW-UP TEST

The follow-up test for the joint Georgia Tech-Tokyo Tech students (phase III of the education project) is shown here. The test consisted of the three questions shown in the following pages. The answers to the questions are:

Question #	Answer
1.	b
2.	d
3.	b

Question #1:

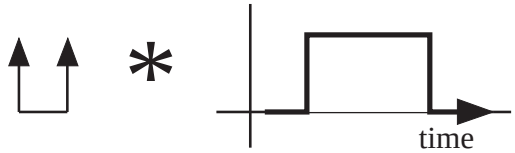
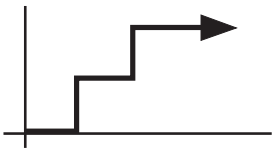
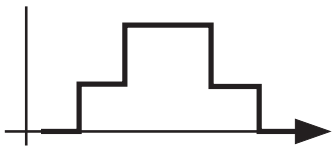


Figure 1.

Which of the following could be the result of convolution operation shown in Figure 1 above:

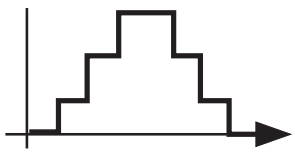


(a)



(b)

(c)



(d)

Question #2:

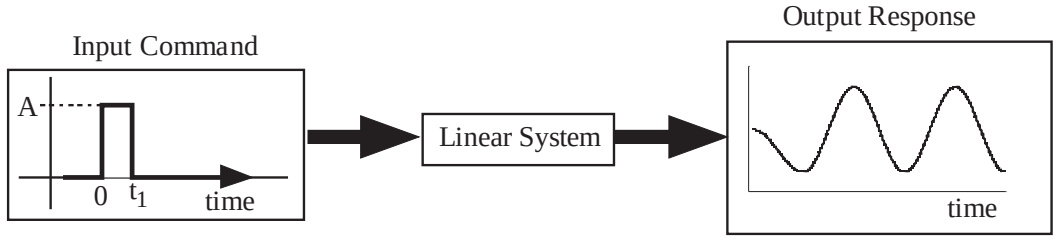


Figure 2

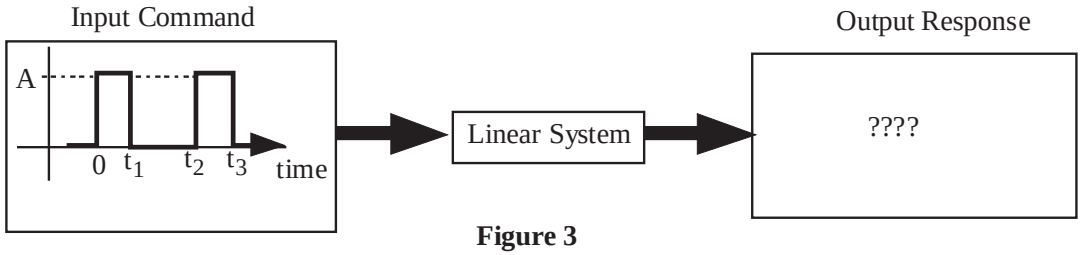


Figure 3

Figure 2 shows the input and output of a linear system. The input command is a pulse with duration t_1 and amplitude A. The output is sinusoidal. Now consider the 2-pulse command shown in Figure 3. The linear system in Figure 3 is the same as the linear system in Figure 2. Which of the following is most accurate about the output response for Figure 3:

a) The residual vibration amplitude could be the same as in Figure 2.
b) The residual vibration amplitude could be greater than Figure 2.
c) The residual vibration amplitude could be less than Figure 2.
d) The residual vibration amplitude could be greater or less than Figure 2.

Question #3:

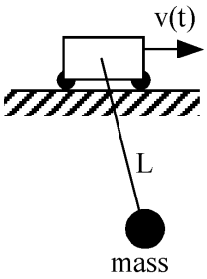


Figure 4. Planar Crane

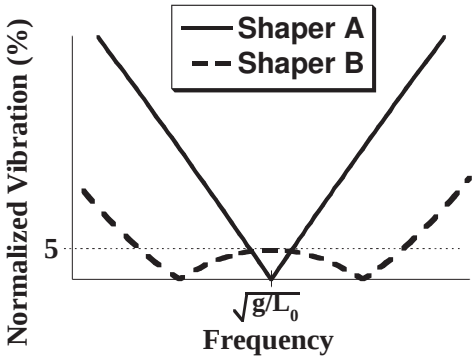


Figure 5. Sensitivity Curves

Suppose we want to move the planar crane in Figure 4 with as little vibration as possible. The hoisted length L changes but its average value is given by L_0 and the natural frequency at this length is given by $\sqrt{g/L_0}$. The velocity command, v(t), is going to be an input shaped command. Figure 5 shows the sensitivity curves for two shapers, shaper A and shaper B, that might be used to move the crane. The horizontal axis shows the frequency and the vertical axis shows the normalized residual vibration for each shaper. Given this information what would you recommend:

a) Use shaper A.
b) Use shaper B.
c) Input shaping cannot be used on this system because L changes.

REFERENCES

[1] Abdel-Rahman, E. M., Nayfeh, A. H., and Masoud, Z. N., "Dynamics and control of cranes: A review," *J. of Vibration and Control*, vol. 9, no. 7, pp. 863–908, 2003.

[2] Agostini, M., Parker, G. G., Groom, K., Schaub, H., and Robinett, R. D., "Command shaping and closed-loop control interactions for a ship crane," in *American Control Conference*, (Anchorage, AK), pp. 2298–2304, 2002.

[3] Al-Garni, A. Z., Moustafa, K. A. F., and Nizami, S. S. A. K. J., "Optimal control of overhead cranes," *Control Engineering Practice*, vol. 3, no. 9, pp. 1277–1284, 1995.

[4] Alli, H. and Singh, T., "Passive control of overhead cranes," *J. of Vibration and Control*, vol. 5, pp. 443–459, 1999.

[5] Auernig, J. W. and Troger, H., "Time optimal control of overhead cranes with hoisting of the load," *Automatica*, vol. 23, no. July, pp. 437–446, 1987.

[6] Beazel, V. M. and Meckl, P. H., "Command shaping applied to nonlinear systems with configuration-dependent resonance," in *American Controls Conference*, (Portland, OR), pp. 539–544, 2005.

[7] Bhat, S. and Miu, D., "Precise point-to-point positioning control of flexible structures," *J. of Dynamic Sys., Meas., and Control*, vol. 112, no. 4, pp. 667–674, 1990.

[8] Blackburn, D. F., Singhose, W., Kitchen, J. P., Petrangenaru, V. P., Lawrence, J., Kamoi, T., and Taura, A., "Advanced input shaping algorithm for nonlinear tower crane dynamics," in *8th International Conference on Motion and Vibration Control*, (Daejeon, Korea), 2006.

[9] Bloom, B. S., *Taxonomy of Educational Objectives, Handbook I.* New York, N. Y.: David McKay Co. Inc., 1956.

[10] Bodson, M., "Experimental comparison of two input shaping methods for the control of resonant systems," in *IFAC World Congress*, (San Francisco, CA), 1996.

[11] Bodson, M., "An adaptive algorithm for the tuning of two input shaping methods," *Automatica*, vol. 34, no. 6, pp. 771–776, 1998.

[12] Bourne, J., Harris, D., and Mayadas, F., "Online engineering education: Learning anywhere, anytime," *J. of Engineering Education*, pp. 131–146, 2005.

[13] Bradley, T. H., Hall, T., Quilin, X., Singhose, W., and Lawrence, J., "Input shaping for nonlinear drive systems," in *ASME International Mechanical Engineering Congress and Exposition*, (Chicago, IL), 2006.

[14] Butler, H., Honderd, G., and Amerongen, J. V., "Model reference adaptive control of a gantry crane scale model," *IEEE Control Systems*, vol. 11, no. January, pp. 57–62, 1991.

[15] Crain, E. A., Singhose, W., and Seering, W. P., "Evaluation of input shaping on configuration dependent systems," in *Japan-USA Symposium of Flexible Automation*, (Boston, MA), pp. 315–318, 1996.

[16] Cutforth, C. F. and Pao, L. Y., "Analysis and design of an adaptive input shaper for the control of flexible structures," in *American Control Conference*, (Anchorage, AK), pp. 1903–1910, 2002.

[17] Dharne, A. G. and Jayasuriya, S., "Increasing the robustness of the input-shaping method using adaptive control," in *American Control Conference*, (Denver, Colorado), pp. 1578–1583, 2003.

[18] Eloundou, R. and Singhose, W., "Interpretation of smooth reference commands as input shaped functions," in *American Control Conference*, (Anchorage, AK), pp. 4948–4953, 2002.

[19] Eloundou, R. and Singhose, W., "Saturation compensating input shapers for reducing vibration," in *6th International Conference on Motion and Vibration Control*, (Saitama, Japan), 2002.

[20] Feisel, L. D. and Rosa, A. J., "The role of the laboratory in undergraeduate engineering education," *J. of Engineering Education*, pp. 121–130, 2005.

[21] Forest, C., Frakes, D., and Singhose, W., "Input-shaped control of grantry cranes: Simulation and curriculum development," in *ASME DETC 18th Biennial Conference on Mechanical Vibration and Noise*, (Pittsburgh, PA), pp. DETC2001/VIB–21522, 2001.

[22] Ginsberg, J. H., *Advanced Engineering Dynamics*. New York: Cambridge University Press, 1998.

[23] Golafshani, A. R. and Aplevich, J. D., "Computation of time-optimal trajectories for tower cranes," in *IEEE Conf. on Control Apps.*, pp. 1134–1139, 1995.

[24] Grose, T. K., "Can distance eduation be unlocked?," *Prism*, vol. 12, no. 8, 2003.

[25] Hekman, K. and Singhose, W., "Suppression of crane payload oscillation using on-off commands," in *American Control Conference*, (Minneapolis, MN), 2006.

[26] Hekman, K., Singhose, W., and Lawrence, J., "Input shaping with coulomb friction compensation on a solder cell machine," in *American Control Conference*, vol. 1, pp. 728–733, 2004.

[27] Hillsley, K. L. and Yurkovich, S., "Vibration control of a two-link flexible robot arm," in *IEEE International Conference on Robotics and Automation*, (Sacrimento, CA), pp. 2121–2126, 1991.

[28] Hong, K.-T. and Hong, K.-S., "Input shaping and vsc of container cranes," in *IEEE International Conference on Control Applications*, (Taipei, Taiwan), pp. 1570–1575, 2004.

[29] Huey, J., *Dynamics and Vibration Control of Large Area Manipulators*. M.s., Georgia Institute of Technology, 2003.

[30] Huey, J., Fortier, J., Wolff, S., Singhose, W., Haraldsson, H. B., Sasaki, S. K., Watari, E., and Lawrence, J., "Remote manipulation of cranes via the internet," in *The 8th International Conference on Motion and Vibration Control*, (Daejeon, Korea), 2006.

[31] Kenison, M. and Singhose, W., "Concurrent design of input shaping and proportional plus derivative feedback control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 124, pp. 398–405, 2002.

[32] Kenison, M. and Singhose, W., "Input shaper design for double-pendulum planar gantry cranes," in *IEEE Conference on Control Applications*, (Hawaii), 1999.

[33] Khalid, A., Singhose, W., Huey, J., and Lawrence, J., "Study of operator behavior, learning, and performance using an input-shaped bridge crane," in *IEEE International Conference on Control Applications*, vol. 1, pp. 759–64, 2004.

[34] Khalid, A., Singhose, W., Huey, J., Lawrence, J., and Frakes, D., "Human operator performance testing using an input shaped bridge crane," *J. of Dynamic Systems, Measurement, and Control*, 2006.

[35] Khorrami, F., Jain, S., and Tzes, A., "Adaptive nonlinear control and input preshaping for flexible-link manipulators," in *American Control Conference*, (San Francisco, CA), pp. 2705–2709, 1993.

[36] Khorrami, F., Jain, S., and Tzes, A., "Experiments on rigid-body based controllers with input preshaping for a two-link flexible manipulator," in *IEEE Transactions on Robotic Automation*, vol. 10, pp. 55–65, 1994.

[37] Khorrami, F., Jain, S., and Tzes, A., "Experimental results on adaptive nonlinear control and input presahping for multi-link flexible manipulators," *Automatica*, vol. 31, no. 1, pp. 83–97, 1995.

[38] Kim, D. and Singhose, W., "Reduction of double pendulum crane oscillations," in *International Conference on Motion and Vibration Control*, (Daejeon, Korea), 2006.

[39] Kinceler, R. and Meckl, P. H., "Corrective input shaping for a flexible-joint manipulator," in *American Control Conference*, (Albuquerque, New Mexico), pp. 1335–1339, 1997.

[40] Kinceler, R. and Peter, M., "Input shaping for nonlinear systems," in *American Controls Conference*, vol. 1, (Seattle, Washington), pp. 914–918, 1995.

[41] Kojima, H. and Singhose, W., "Adaptive deflection limiting control for slewing flexible space structures," in *56th International Astronautical Congress*, vol. IAC-05-C2.7.02, (Fukuoka, Japan), 2005.

[42] Kojima, H. and Singhose, W., "Adaptive deflection limiting control for slewing flexible space structures," *AIAA J. of Guidance, Control, and Dynamics*, Accepted 2006.

[43] Kozak, K., *Robust Command Generation for Nonlinear Systems.* Ph.d., Georgia Institute of Technology, 2003.

[44] Kozak, K., Ebert-Uphoff, I., and Singhose, W., "Analysis of varying natural frequencies and damping ratios of a sample parallel manipulator throughout its workspace using linearized equations of motion," in *ASME Design Engineering Technical Conference*, (Pittsburgh, PA), pp. 1935–1941, 2001.

[45] Kozak, K., Ebert-Uphoff, I., and Singhose, W., "Locally linearized dynamic analysis of parallel manipulators and application of input shaping to reduce vibrations," *ASME J. of Mechanical Design*, vol. 126, pp. 156–168, 2004.

[46] Kress, R. L., Fansen, J. F., and Noakes, M. W., "Experimental implementationof a robust damped-oscillation control algorithm on a full sized, two-dof, ac induction motor-driven crane," in *5th International Symposium on Robotics and Manufacturing*, pp. 585–92, 1994.

[47] Lawrence, J., Danielson, J., and Singhose, W., "Design and analysis of input shapers for systems with a braking nonlinearity," in *International Symposium on Flexible Automation*, (Osaka, Japan), 2006.

[48] Lawrence, J., Fatkin, B., Singhose, W., Weiss, R., Erb, A., and Glauser, U., "An internet-driven tower crane for dynamics and controls education," in *7th IFAC Symposium on Advances in Control Education*, (Madrid, Spain), 2006.

[49] Lawrence, J., Singhose, W., and Hekman, K., "Friction-compensating command shaping for vibration reduction," *ASME J. of Vibration and Acoustics*, vol. 127, pp. 307–314, 2005.

[50] Lawrence, J., Falkenberg, M., Singhose, W., and Pelaez, G., "Input shaping for a flexible, nonlinear, one-link robotic arm with backlash," in *2004 USA-JAPAN Symposium on Flexible Automation*, (Denver, Colorado), ASME, 2004.

[51] Lawrence, J. and Singhose, W., "An analytical solution for a zero vibration input shaper for systems with coulomb friction," *American Control Conference*, 2002.

[52] Lawrence, J. and Singhose, W., "Decreasing effects of coulomb friction in precision positioning using input shaping: Experimental verifications," in *Japan-USA Symp. on Flexible Automation*, (Hiroshima, Japan), 2002.

[53] Lawrence, J. and Singhose, W., "Design of a minicrane for education and research," in *6th International Workshop on Research and Education in Mechatronics*, (Annecy, France), pp. 254–259, 2005.

[54] Lawrence, J., Singhose, W., and Heckman, K., "Robust friction-compensating input shapers," in *8th Cairo Univ. Int. Conf. on Mechanical Design and Production*, (Cairo, Egypt), 2004.

[55] Lewis, D., Parker, G. G., Driessen, B., and Robinett, R. D., "Command shapiong control of an operator-in-the-loop boom crane," in *American Control Conference*, (Philadelphia, PA), pp. 2643–7, 1998.

[56] Lewis, D., Parker, G. G., Driessen, B., and Robinett, R. D., "Comparison of command shaping controllers for supressing payload sway in a rotary boom crane," in *IEEE Internation Conference on Control Applications*, (Kohala Coast-Island of Hawai'i, Hawai'i), pp. 719–724, 1999.

[57] Leydens, J. A., Moskal, B. M., and Pavelich, M. J., "Qualitative methods used in the assessment of engineering education," *J. of Engineering Education*, 2004.

[58] Lim, S. and How, J., "Input command shaping techniques for robust, high-performance control of flexible structures," in *AIAA Guidance, Navigation, and Control Conf.*, (San Diego, CA), 1996.

[59] Lim, S., Stevens, H. D., and How, J., "Input shaping design for multi-input flexible systems," *J. of Dynamic Systems, Measurement, and Control*, vol. 121, pp. 443–7, 1999.

[60] Magee, D. P. and Book, W. J., "The application of input shaping to a system with varying parameters," in *Japan/USA Conference on Flexible Automation*, pp. 519–526, 1992.

[61] Magee, D. P. and Book, W. J., "Eliminating multiple modes of vibration in a flexible manipulator," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 474 – 479, 1993.

[62] Magee, D. P. and Book, W. J., "Implementing modified command filtering to eliminate multiple modes of vibration," in *American Controls Conference*, vol. 3, (San Francisco, CA), pp. 2700–2704, 1993.

[63] Moustafa, K. A. F. and Ebeid, A. M., "Nonlinear modeling and control of overhead crane load sway," *J. of Dynamic Systems, Measurement, and Control*, vol. 110, pp. 266–271, 1988.

[64] Nise, N. S., *Control Systems Engineering*. 2nd ed., 1995.

[65] Noakes, M. W. and Jansen, J. F., "Generalized inputs for damped-vibration control of suspended payloads," *Robotics and Autonomous Systems*, vol. 10, no. 2, pp. 199–205, 1992.

[66] O'Connor, W. J., "A gantry crane problem solved," *J. of Dynamic Systems, Measurement, and Control*, vol. 125, pp. 569–576, 2003.

[67] Omar, H. M. and Nayfeh, A. H., "Gain scheduling feedback control for tower cranes," *J. of Vibration and Control*, vol. 9, pp. 399–418, 2003.

[68] Pahl, G. and Beitz, W., *Engineering Design: A systematic Approach*. Springer, 2001.

[69] Park, B. J., Hong, K.-S., and Huh, C., "Time-efficient input shaping control of container cranes," in *International Conference on Control Applications*, (Anchorage, AK), pp. 80–85, 2000.

[70] Park, J., Chang, P. H., and Lee, E., "Can a time invariant input shaping technique eliminate residual vibrations of LTV systems," in *American Controls Conference*, (Anchorage, AK), pp. 2292–2297, 2002.

[71] Park, J. and Chang, P.-H., "Learning input shaping technique for non-lti systems," in *American Control Conference*, (Philadelphi, Pennsylvania), pp. 2652–2656, 1998.

[72] Peterson, G. D. and Feisel, L. D., "e-learning: The challenge for engineering education," in *e-Technologies in Engineering Education, A United Engineering Foundation Conference*, (Davos, Switzerland), pp. 164–169, 2002.

[73] Rhim, S. and Book, W. J., "Adaptive command shaping using adaptive filter approach in time domain," in *American Control Conference*, (San Diego, CA), pp. 81–85, 1999.

[74] Rhim, S. and Book, W. J., "Noise effect on adaptive command shaping methods for flexible manipulator control," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 1, pp. 84–92, 2001.

[75] Rhim, S., Hu, A.-P., Sadegh, N., and Book, W. J., "Combining a multirate repetitive learning controller with command shaping for improved flexible manipulator control," *J. of Dynamic Systems, Measurement, and Control*, vol. 123, pp. 385–390, 2001.

[76] Robertson, M. and Singhose, W., "Multi-level optimization techniques for designing digital shapers," in *American Control Conference*, (Pittsburgh, PA), 2001.

[77] Robinett, R. I., Feddema, J. T., Eisler, G., WIlson, D., and Parker, G., "Flexible robotic manipulator research at sandia national laboratories," in *11th Symposium on Structural Dynamics and Control*, (Blacksburg, VA), 1997.

[78] Singer, N. and Seering, W. P., "An extension of command shaping methods for controlling residual vibration using frequency sampling," in *IEEE International Conference on Robotics and Automation*, (Nice, France), pp. 800–805, 1992.

[79] Singer, N., Singhose, W., and Kriikku, E., "An input shaping controller enabling cranes to move without sway," in *ANS 7th Topical Meeting on Robotics and Remote Systems*, (Augusta, GA), 1997.

[80] Singer, N. C., "Residual vibration reduction in computer controlled machines," Technical Report MIT Artificial Intelligence Laboratory Technical Report Number AITR-1030, MIT Artificial Intelligence Lab, February, 1989 1989.

[81] Singer, N. C. and Seering, W. P., "Preshaping command inputs to reduce system vibration," *J. of Dynamic Sys., Measurement, and Control*, vol. 112, no. March, pp. 76–82, 1990.

[82] Singh, T., "Jerk limited input shapers," *J. of Dynamic Systems, Measurement, and Control*, vol. 126, pp. 215–219, 2004.

[83] Singhose, P. W., 2006.

[84] Singhose, W., Seering, W. P., and Singer, N., "Shaping inputs to reduce vibration: A vector diagram approach," in *IEEE International Conference on Robotics and Automation*, 1990.

[85] Singhose, W., Seering, W. P., and Singer, N., "Residual vibration reduction using vector diagrams to generate shaped inputs," *J. of Mechanical Design*, vol. 116, pp. 654–659, 1994.

[86] Singhose, W., Huey, J., Lawrence, J., and Frakes, D., "Input shaping curriculum: Integrating interactive simulations and experimental setups," in *11th Mediterranean Conference on Control Automation*, (Rhodes, Greece), 2003.

[87] Singhose, W., Porter, L., Kenison, M., and Kriikku, E., "Effects of hoisting on the input shaping control of gantry cranes," *Control Engineering Practice*, vol. 8, no. 10, pp. 1159–1165, 2000.

[88] Singhose, W., Singer, N., and Seering, W., "Time-optimal negative input shapers," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 119, no. June, pp. 198–205, 1997.

[89] Singhose, W. and Towell, S., "Double-pendelum gantry crane dynamics and control," in *IEEE Conf. on Control Applications*, (Trieste, Italy), 1998.

[90] Singhose, W. E., Porter, L. J., and Seering, W. P., "Input shaped control of a planar gantry crane with hoisting," in *American Control Conf.*, (Albuquerque, NM), 1997.

[91] Slotine, J.-J. and Li, W., *Applied Nonlinear Control*. New Jersey: Prentice-Hall, 1991.

[92] smith, J. Y., Kozak, K., and Singhose, W., "Input shaping for a simple nonlinear system," in *American Control Conference*, (Anchorage, AK), pp. 821–826, 2002.

[93] Smith, O. J. M., *Feedback Control Systems*. New York: McGraw-Hill Book Co., 1958.

[94] Smith, O. J. M., "Posicast control of damped oscillatory systems," *Proceedings of the IRE*, vol. 45, no. September, pp. 1249–1255, 1957.

[95] Sorensen, K., Singhose, W., and Dickerson, S., "A controller enabling precise positioning and sway reduction in bridge and gantry cranes," *Control Engineering Practice*, 2006.

[96] Sorensen, K., Singhose, W., and Dickerson, S., "A controller enabling precise positioning and sway reduction in cranes with on-off actuation," in *IFAC*, (Prague), 2005.

[97] Sorensen, K. L., *A Combined Feedback and Command Shaping Controller for Improving Positioningand Reducing Cable Sway in Cranes*. Masters, Georgia Institute of Technology, 2005.

[98] Starr, G. P., "Swing-free transport of suspendedobjects with a path-controlled robot manipulator," *J. of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 97–100, 1985.

[99] Stergiopoulos, J., Manesis, S., Tzes, A., and Nikolakopoulos, G., "Control via input shaping of a pneumatic crane system," in *American Control Conference*, pp. 545 – 550, 2005.

[100] Stergiopoulos, J. and Tzes, A., "Adaptive input shaping for nonlinear systems: A case study," in *Proceedings of the 13th Mediterranean Conference on Control and Automation*, (Limassol, Cyprus), pp. 188–193, 2005.

[101] Trefethen, L. N., *Spectral Methods in Matlab*. Philadelphia, PA: SIAM, 2000.

[102] Tzes, A. and Yurkovich, S., "An adaptive input shaping control scheme for vibration suppression in slewing flexible structures," *IEEE Transactions on Control Systems Technology*, vol. 1, no. June, pp. 114–121, 1993.